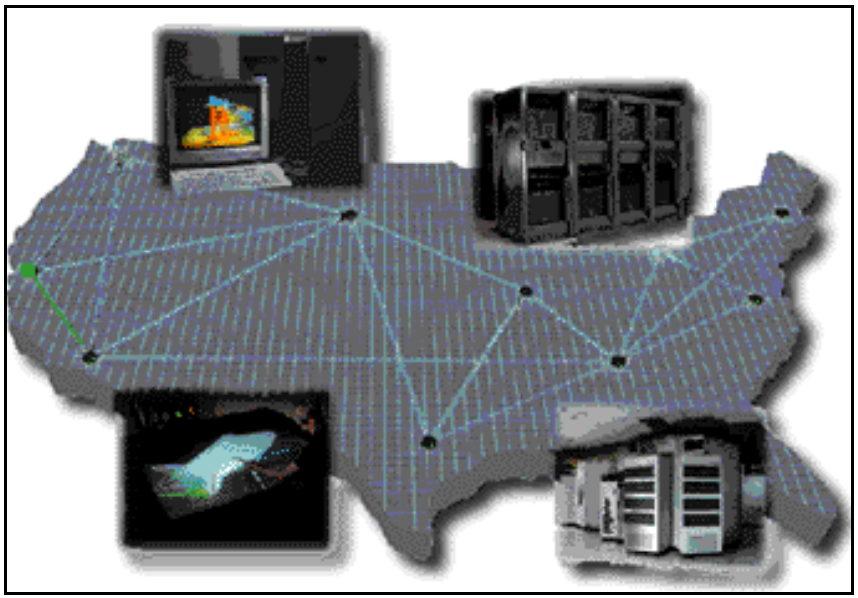
Information Power Grid:

Distributed High-Performance Computing, Large-Scale Data Management, and Collaboration Environments for Science and Engineering

William E. Johnston, Dennis Gannon, William Nitzberg, William Van Dalsem



Numerical Aerospace Simulation Facility at NASA Ames Research Center

William J. Feiereisen, Division Chief

http://nas.nasa.gov/~wej/home/IPG

This talk has three major sections:

I: Motivation for “grid” computing and information environments

II: The approach for building the Information Power Grid as an instance of such an environment

III: IPG implementation plans and status

Part I: Motivation

Prototype applications have demonstrated both the potential and the "reality" of high-speed, service and resource based, wide area distributed systems.

So ....

What new science and engineering can be done when applications can routinely access remote resources via high-speed networks?

How can we facilitate this with this with computational and data grids like IPG?

- **Two real examples and one gedankenexperiment**

Example: Real-Time Digital Libraries for On-Line, High Data-Rate Instruments [3]

Goal was to provide cardiac care physicians with immediate access to major patient studies done on remote imaging systems, rather that having to wait for weeks for the paper report.

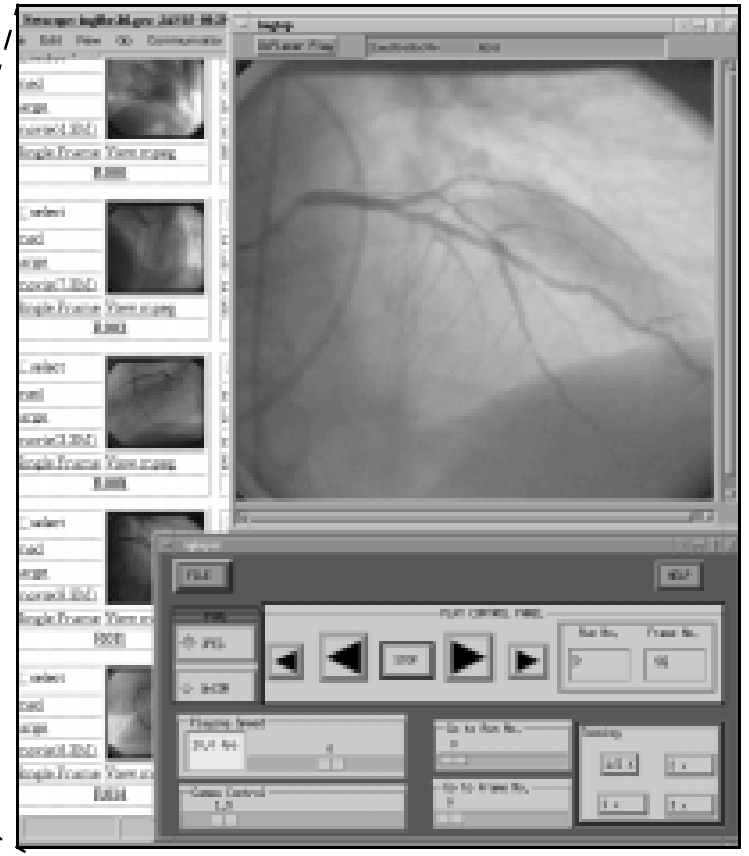
Demonstrated technologies:

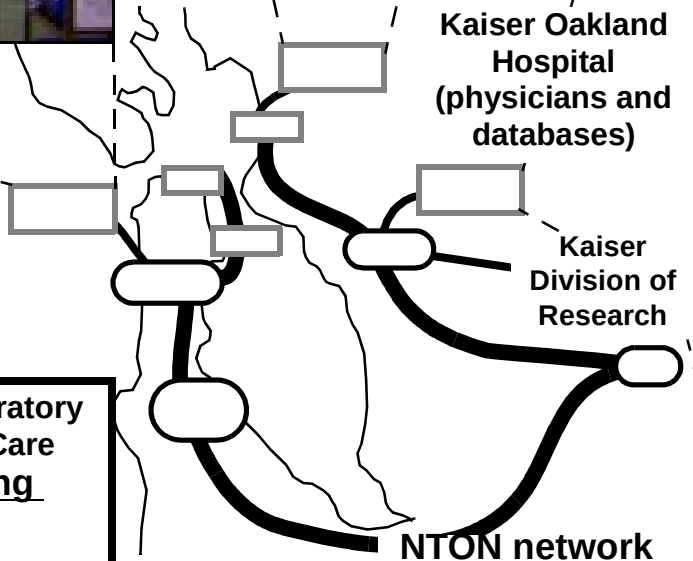
- **On-line, real-time, high data-rate instrument**
- **Management of large data sets in wide area**
- **Optical WDM metropolitan area network (now part of NGI)**
- **Remote data analysis followed by automatic data cataloguing and archiving**
- **Remote data users**
- **Widely distributed, high performance "application-level" cache**
- **A "data flow" architecture for high data-rate on-line instrumentation systems**

WALDO real-time digital library system and DPSS distributed cache [4] for data cataloguing and storage

Compute servers for data analysis and transformation

Kaiser San Francisco Hospital Cardiac Catheterization Lab (X-ray video imaging system, ≈ 130 mbit/s, 50% duty cycle 8-10 hr/day)



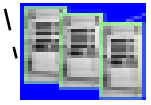


Kaiser Oakland Hospital (physicians and databases)

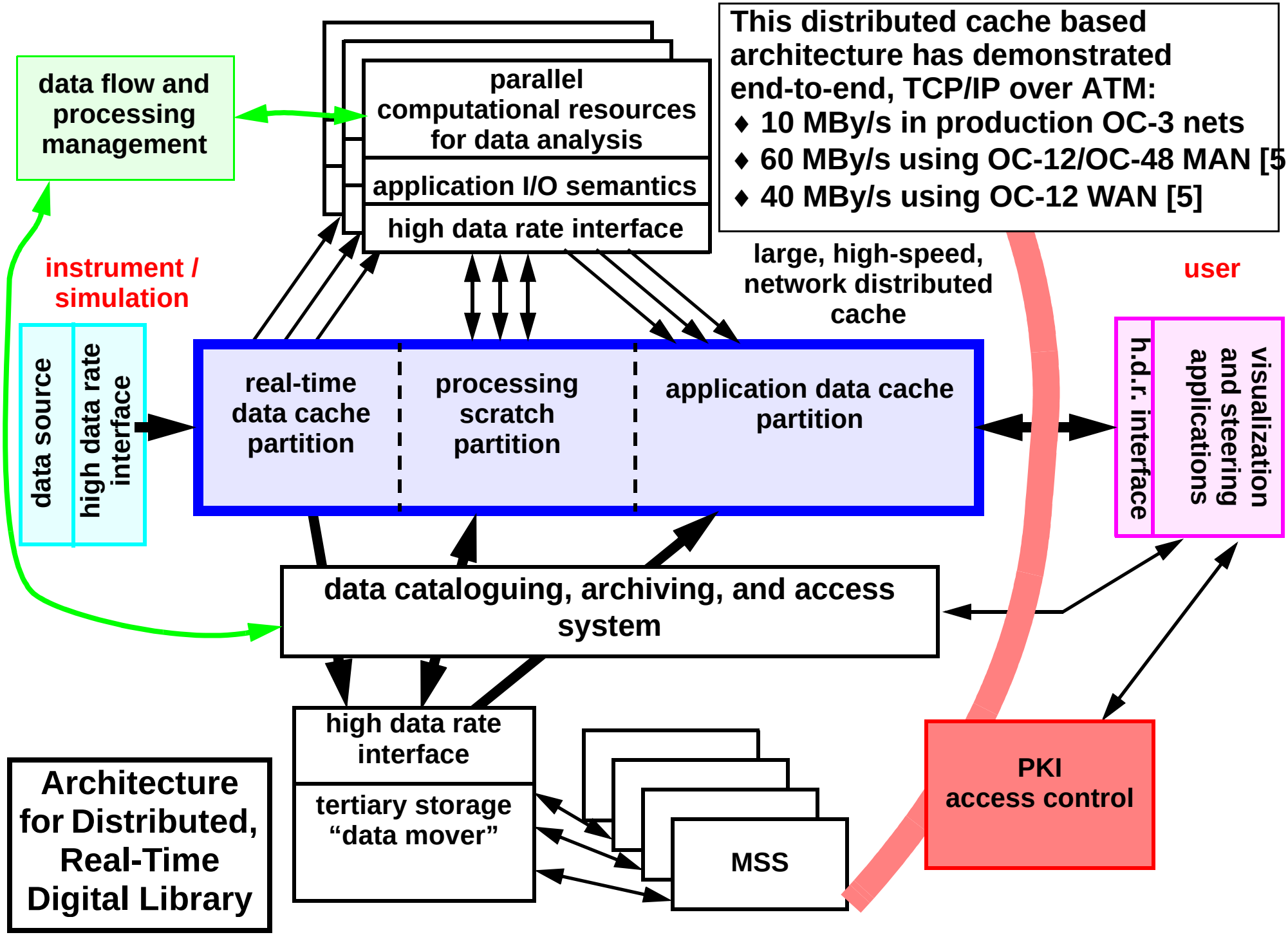
Kaiser Division of Research

NTON network testbed

The PSE: Automatically generated user interfaces providing indexed access to the large data objects (the X-ray video) and to various derived data.



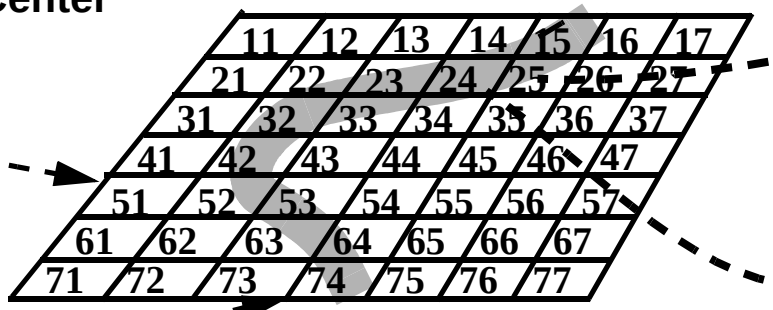
Lawrence Berkeley National Laboratory and Kaiser Permanente Health Care
On-line Health Care Imaging Experiment
in the San Francisco Bay Area


This distributed cache based architecture has demonstrated end-to-end, TCP/IP over ATM:
♦ 10 MBy/s in production OC-3 nets
♦ 60 MBy/s using OC-12/OC-48 MAN [5
♦ 40 MBy/s using OC-12 WAN [5]
data flow and processing management
parallel computational resources for data analysis
application I/O semantics
high data rate interface
instrument / simulation
large, high-speed, network distributed cache
user
data source
high data rate interface
real-time data cache partition
processing scratch partition
application data cache partition
h.d.r. interface
visualization and steering applications
data cataloguing, archiving, and access system
high data rate interface
tertiary storage "data mover"
MSS
PKI access control
Architecture for Distributed, Real-Time Digital Library

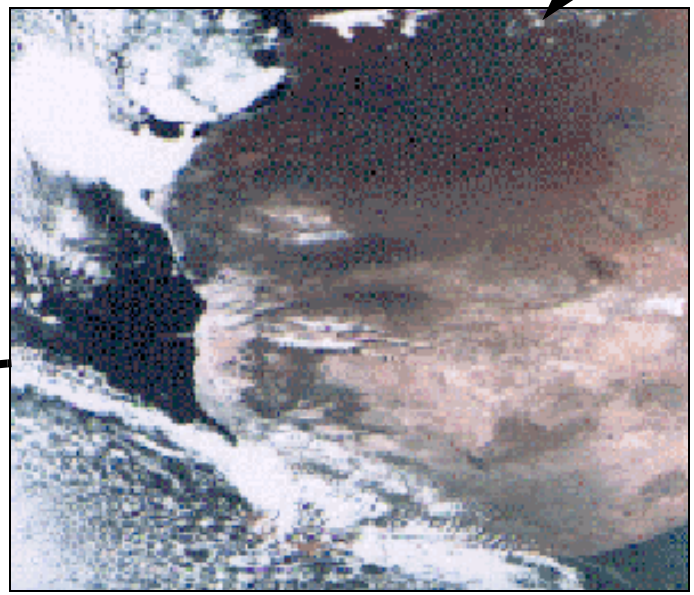
Example: High Data-Rate Distributed Data Management and Federated Access for Satellite and Aerial Imagery, Digital Terrain Data, and Atmospheric Data [6]

- **On-line, real-time access to multiple environmental data sets that are (and always will be) maintained by domain experts at their own sites.**
- **On demand, real-time interactive exploration of an operational environment (military, community emergency services)**
- **Aggregation of multiple, widely distributed, multi-discipline data sets**
- **DARPA MAGIC testbed consortium (see www.magic.net) developed distributed services, data and visualization from EROS Data Center, NCAR, NAVO, SRI (collab. with NASA NREN)**
- **MAGIC wide-area, gigabit network testbed is now part of NGI**

Landscape represented by tiled images and terrain at EROS Data Center



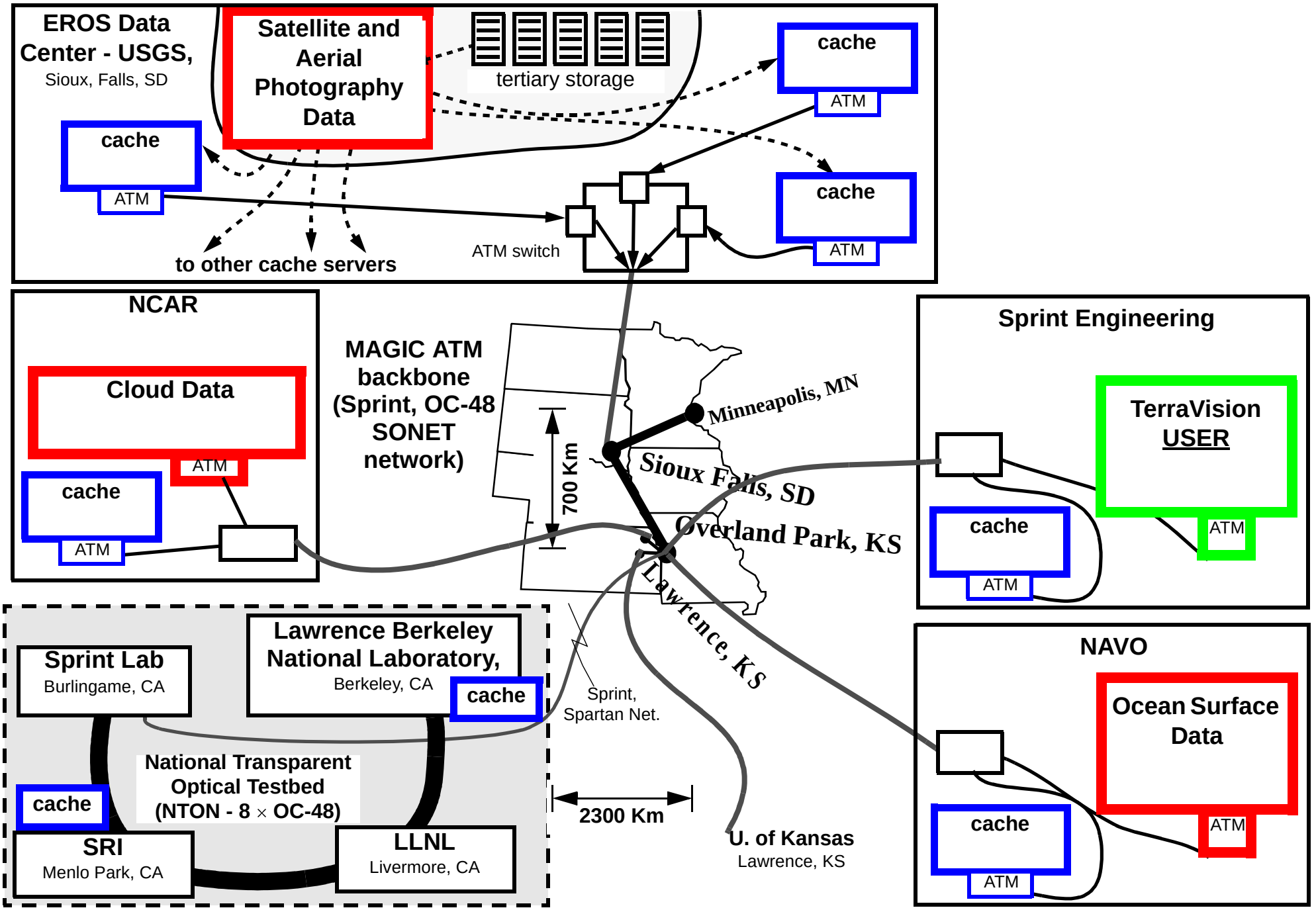
Path of travel



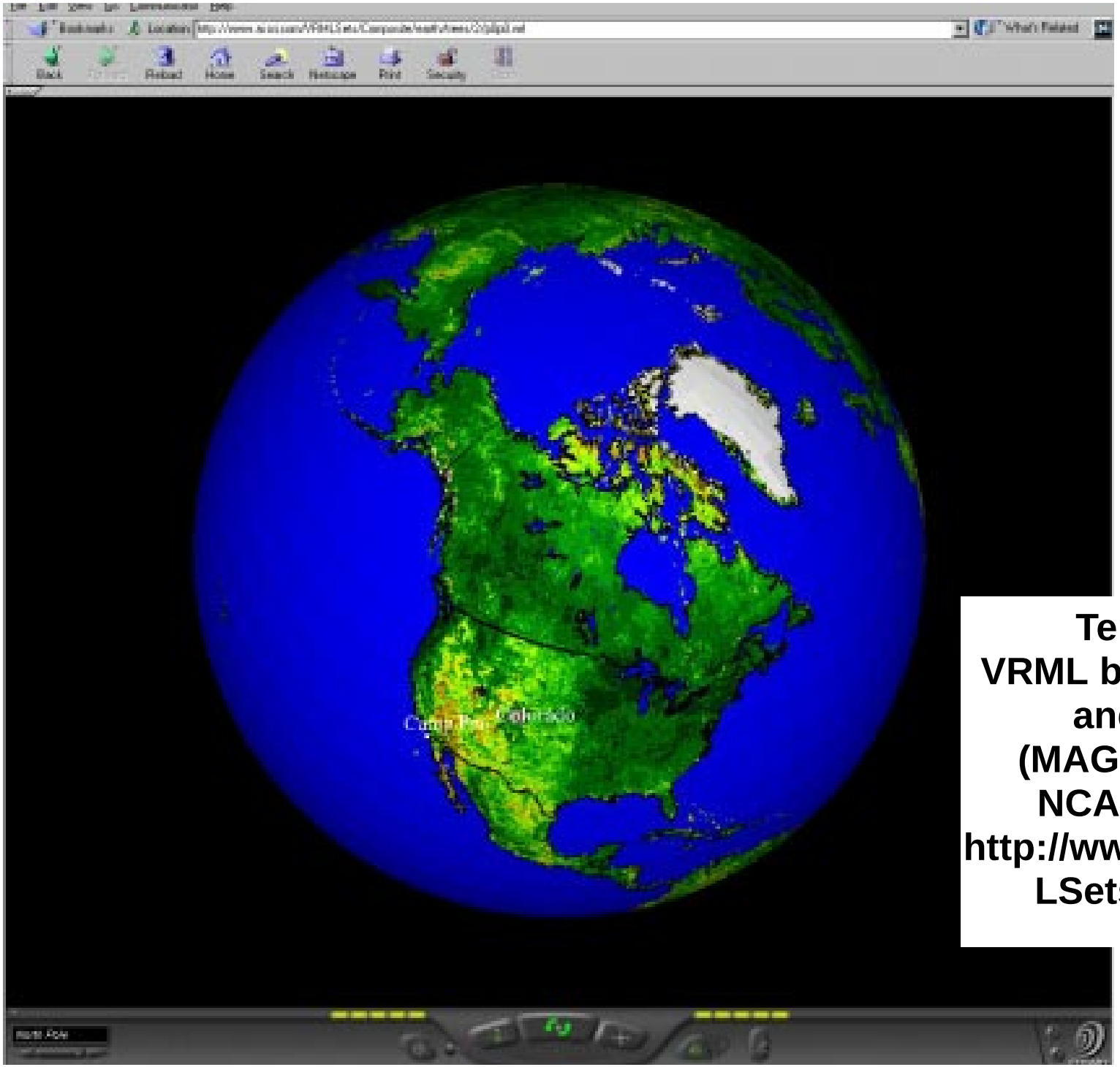
TerraVision produces a accurate visualization of the landscape

Human user navigates (controls path of travel)

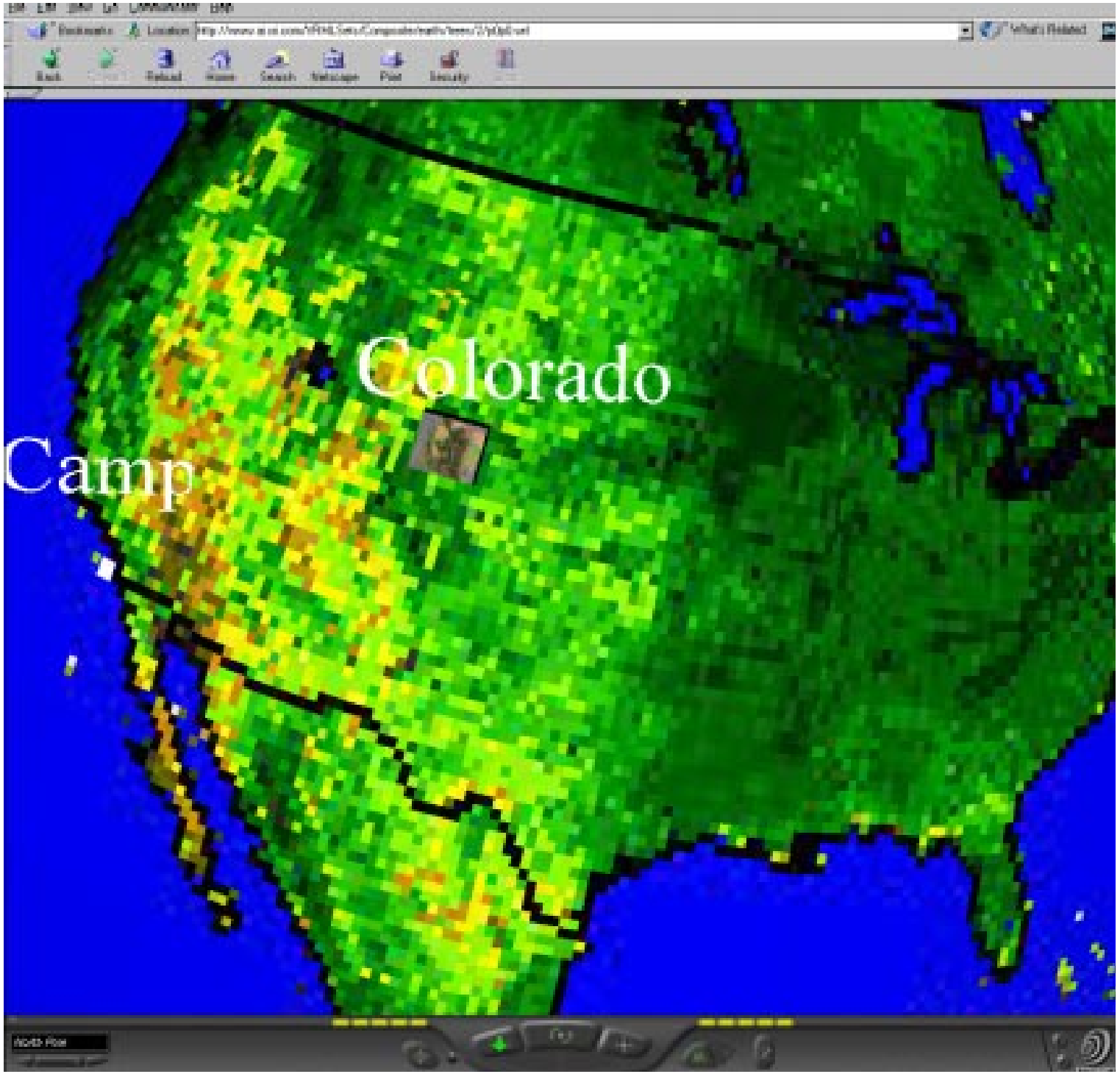
TerraVision Provides Real-time Visualization of Aggregated Data

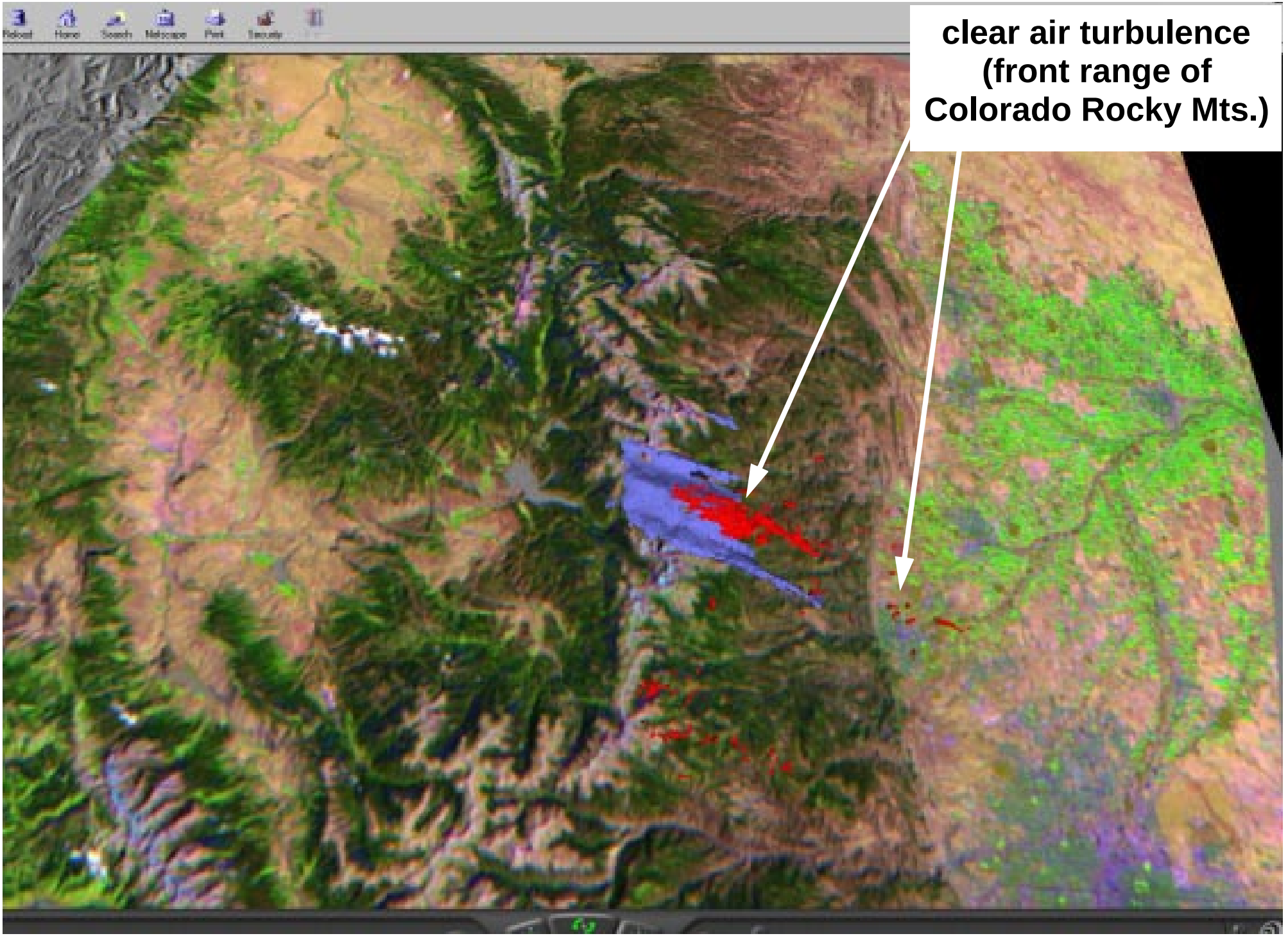


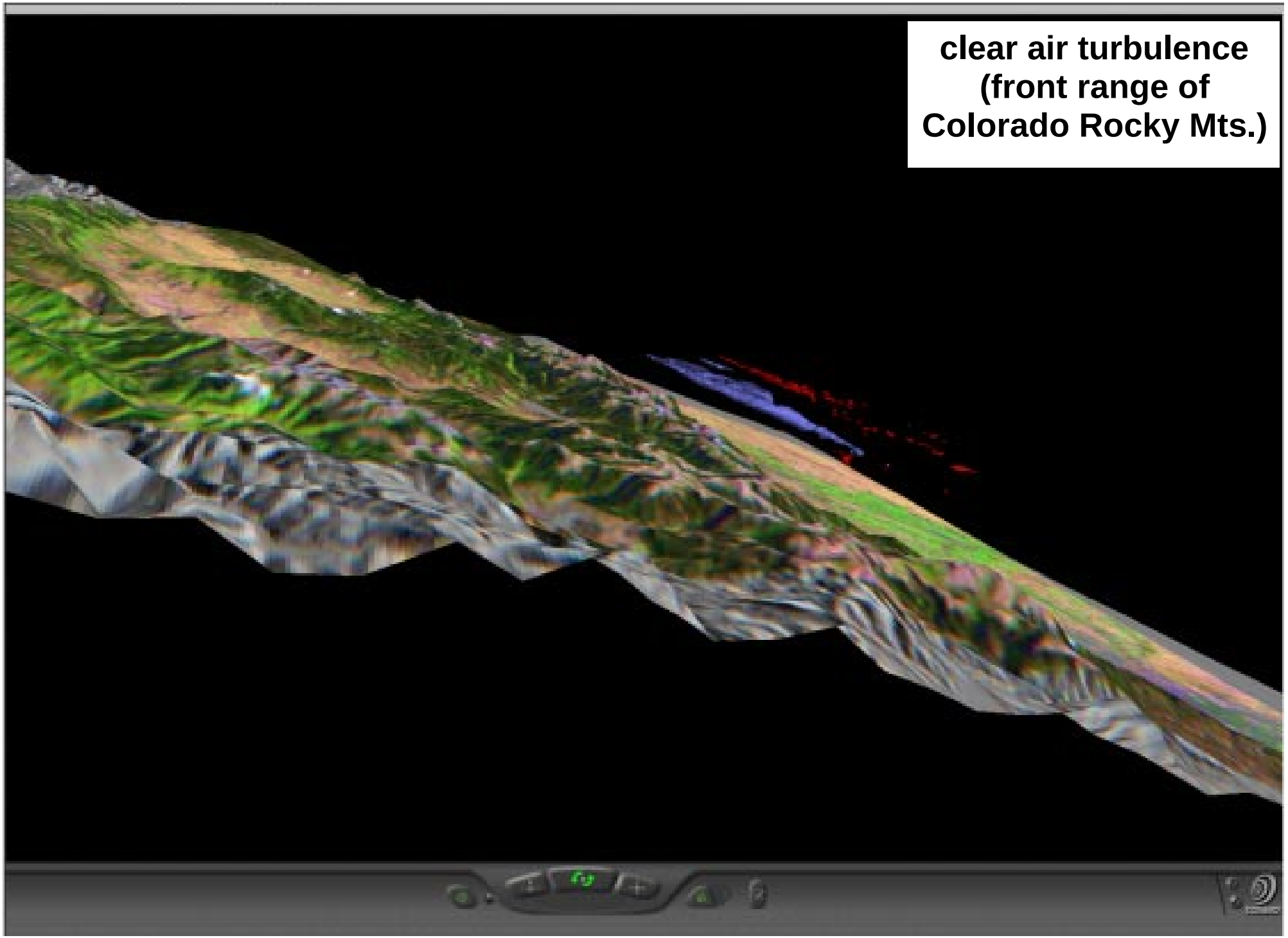
The MAGIC Testbed Distributed Application Environment

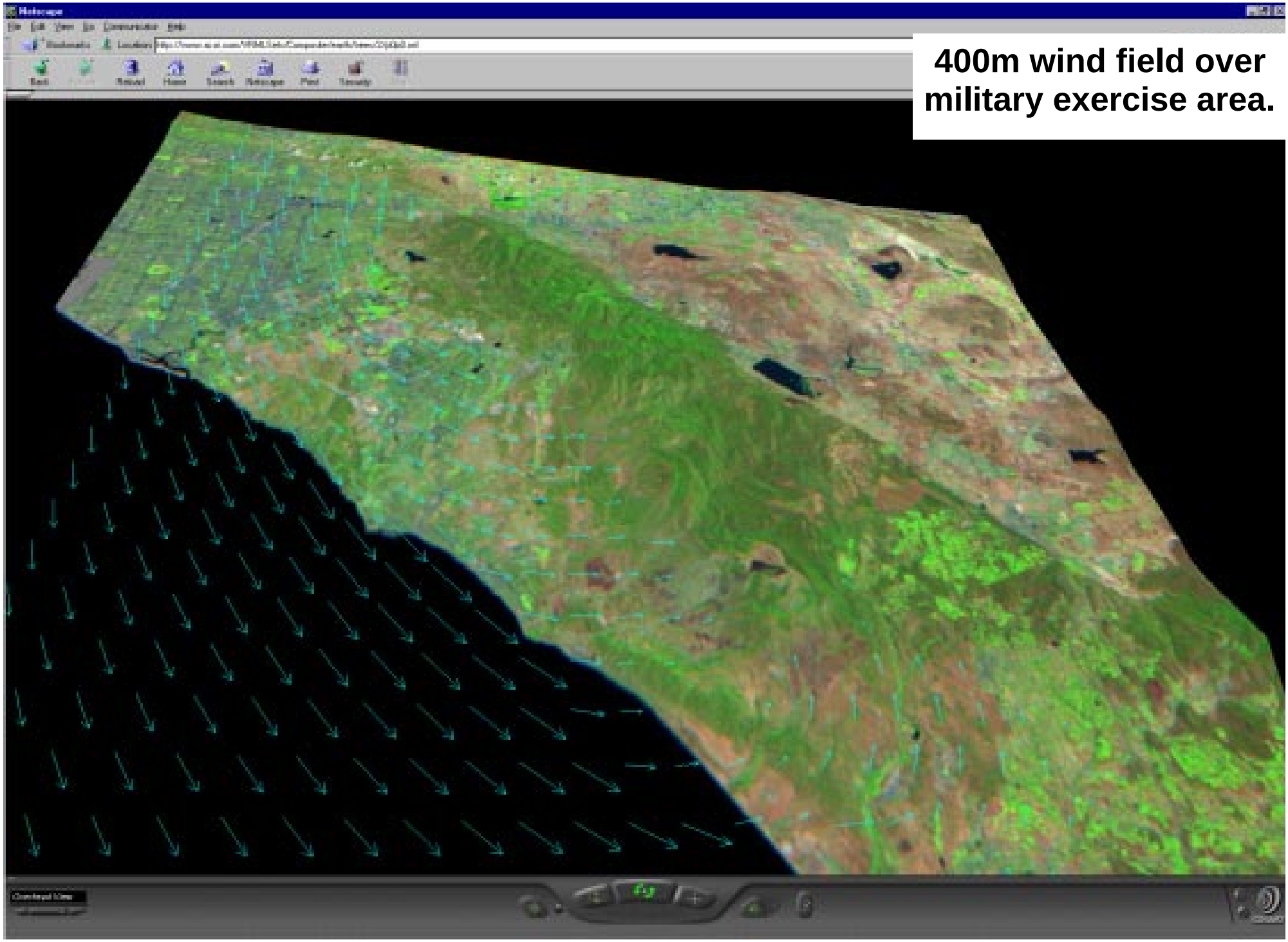


TerraVision-2:
VRML based data fusion and browsing.
(MAGIC consortium, NCAR, and NAVO: http://www.ai.sri.com/VRMLSets/Composite/)


Colorado
Camp


clear air turbulence
(front range of
Colorado Rocky Mts.)


clear air turbulence
(front range of
Colorado Rocky Mts.)


400m wind field over
military exercise area.

Distributed Simulation

Some problems that are too large for a single computing platform that can be distributed "naturally" in a grid environment:

- **Not usually true for single address space / tightly coupled algorithms**

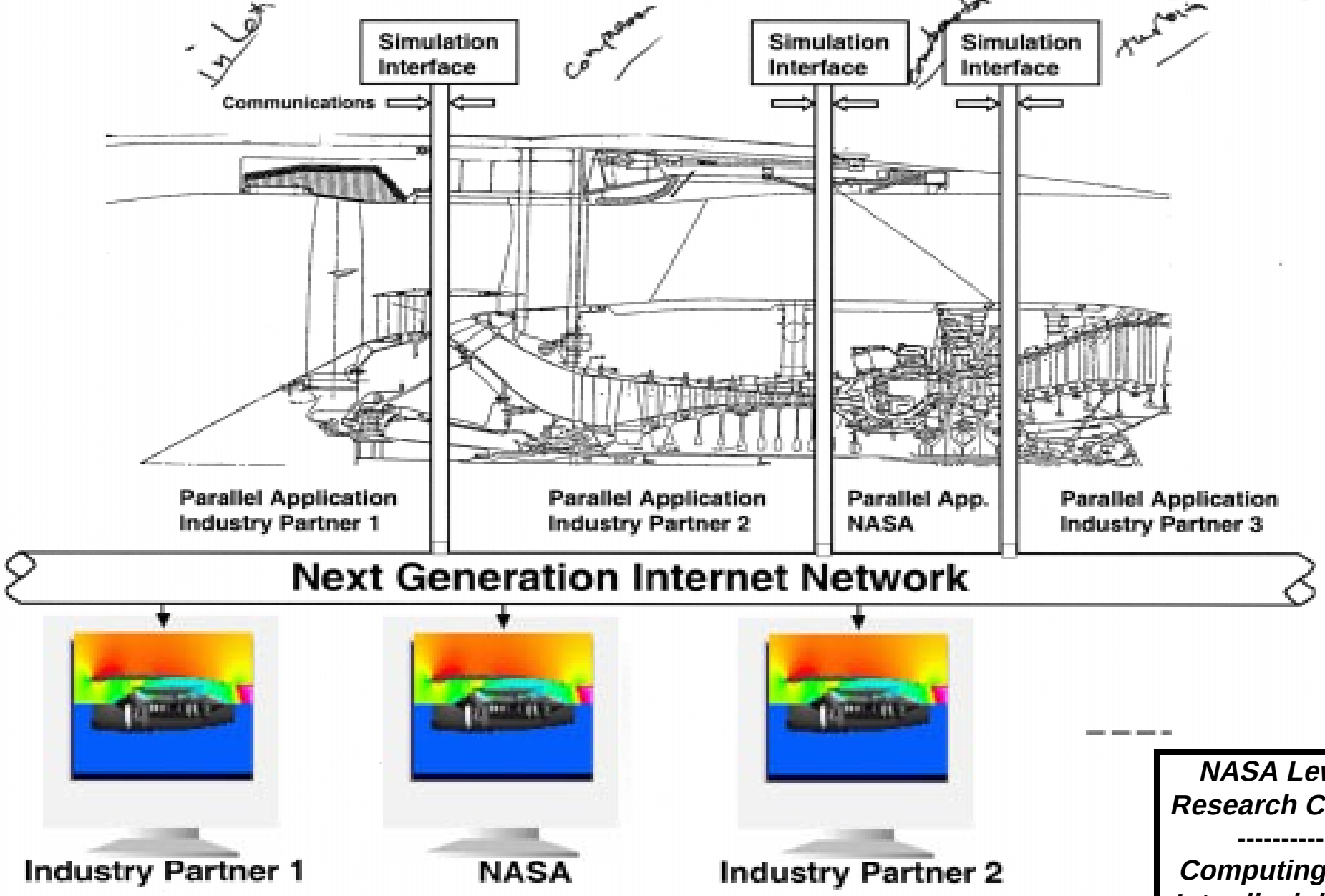
  (What most people call "teraflop computing" will happen on the grid when teraflop computers are connected to the grid as a resource.)

- **True for inherently parallel, loosely coupled problems (event analysis, parameter studies, etc.)**

- **Sometimes true for coupled simulations**

Large-scale, "independently" computing simulation components with "reasonable" communications needs can be combined to provide "whole" system simulations by using the services of the grid.

Distributed Simulation Environment



Full simulation visualized at all locations

NASA Lewis Research Center

Computing and Interdisciplinary Systems Office

What Was Learned From the Examples?

Building High Volume, Distributed Data Management and Analysis Systems

Each application example* depends on widely distributed resources communicating over high-speed networks and had to be designed, developed, and debugged "end-to-end, top-to-bottom":

*Several of these examples are not NASA applications, they all represent - through IPG - various NASA partnerships.

- **end-to-end: the full scope of the application from data generation and management through computation to user interface (all typically remote from each other) must be addressed**
- **top-to-bottom: means that every aspect of the distributed system from data storage/generation and CPU elements, all the way down through the network fabric, must be monitored, evaluated, and refined**

This comprehensive "system approach" is essential for making widely distributed, high-speed applications work, and grid services are intended to facilitate this approach.

Characteristics of Data Intensive Computing

- Wide area data handling
  - manage instrument data streams and/or simulation output
- Cataloguing (preferably automatic)
  - describe data (generate metadata) and data formats
  - assign use conditions
  - publish/subscribe
- Multiple archive management
- Data access
  - dataset “access protocols” (e.g. http, ftp, nfs, ...)
  - uniform I/O mechanisms (e.g. read, write, seek for all access)
  - “discover” data syntax (structure)
- Computational data analysis / transformation on remote systems
- Transience
  - many application systems will use distributed resource configurations that need to be built on-demand and are used for limited periods
- Rich application domain interfaces

Services Required for Distributed Computing and Data Management

- Transparent use and management of federated, multiple, “annotated” data archives
- Readily available cataloguing services
- Data location management
- Rich data access and I/O services
- Fault tolerance and autonomous management of application, system, and infrastructure components
- Resource discovery and brokering, and co-scheduling and reservation

- **Access control (stakeholder management of use-conditions and infrastructure security)**
- **Toolkits for integrating collaborative components**
- **Toolkits for <u>building</u> user interface systems (problem solving environments / workbenches)**

What Kinds of Applications can be Built With These Services?

- **Data intensive computing and instrument access**
  (The focus of this analysis.)
- **Distributed computation systems using aggregated resources**
- **Problem Solving Environments / workbenches**
- **Computer mediated human collaboration**

These capabilities are related in that they:

- **are all needed to support grid based science and engineering**
- **have common requirements for the supporting infrastructure - "grid common services"**

The Services are Intended to Support "Large-Scale" Applications

"Scale" refers to several dimensions:

- **Large scale computational and storage capacity through aggregation of resources**
- **Scale in the complexity of resources (independent of capacity)**
  - **data intensive computing tends to require a complex mix of resources, with or without high capacity**
  - **management of very diverse data is one key issue**
  - **grids are intended to provide transparent access to these resources**

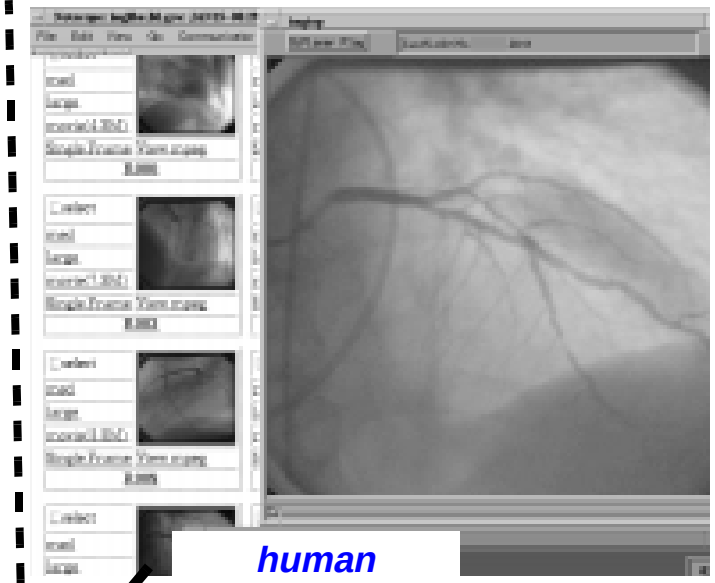
instrument resources

compute resources

resource manager

Net3

resource manager



human collaborators

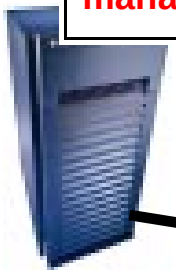
resource manager

Net1

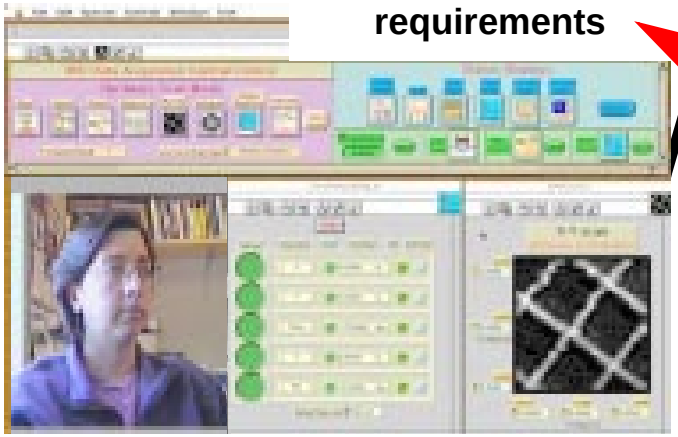
resource manager

resource manager

resource manager



application "system" requirements



resource manager

Net2

resource manager

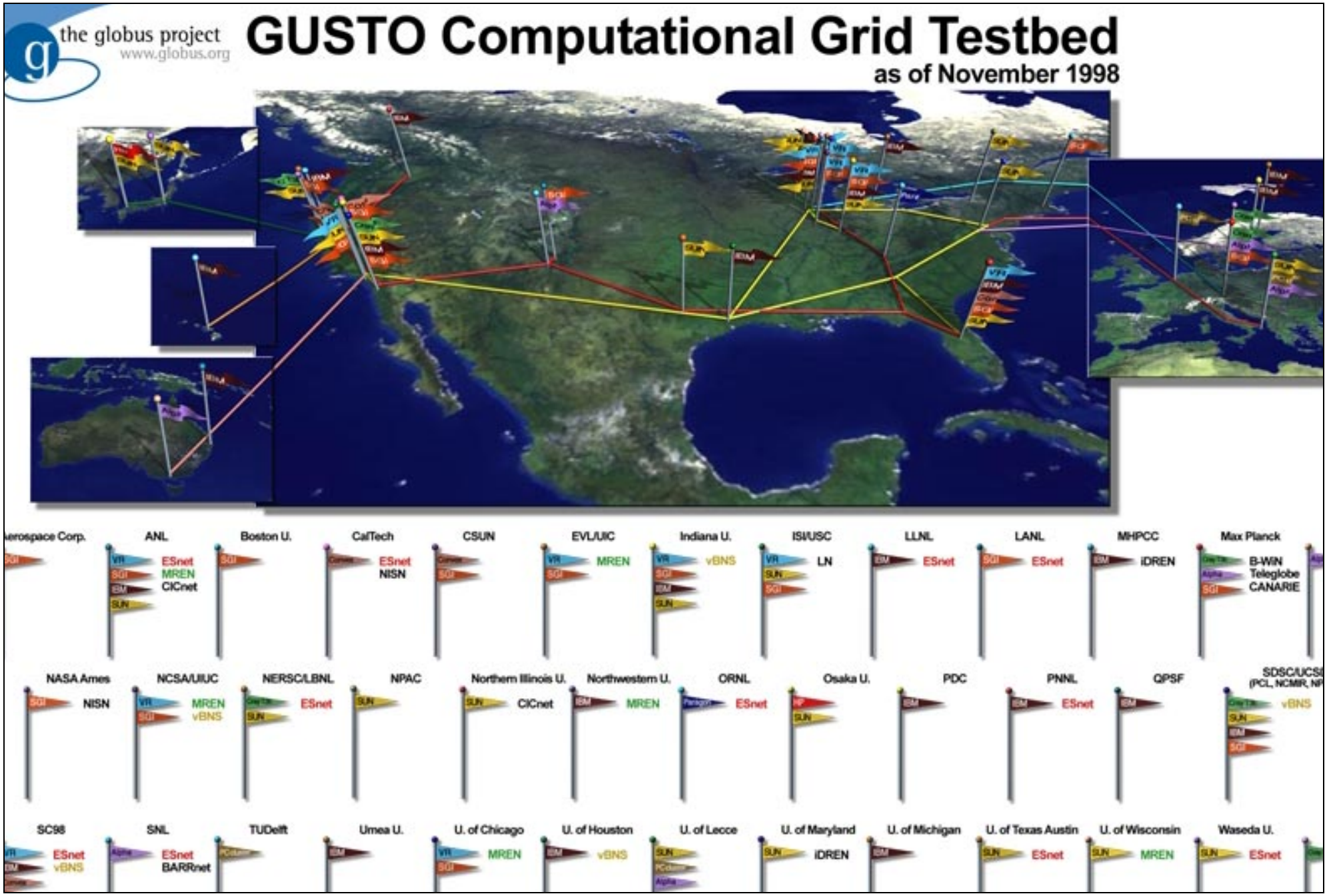
storage resources

resource broker

network bandwidth

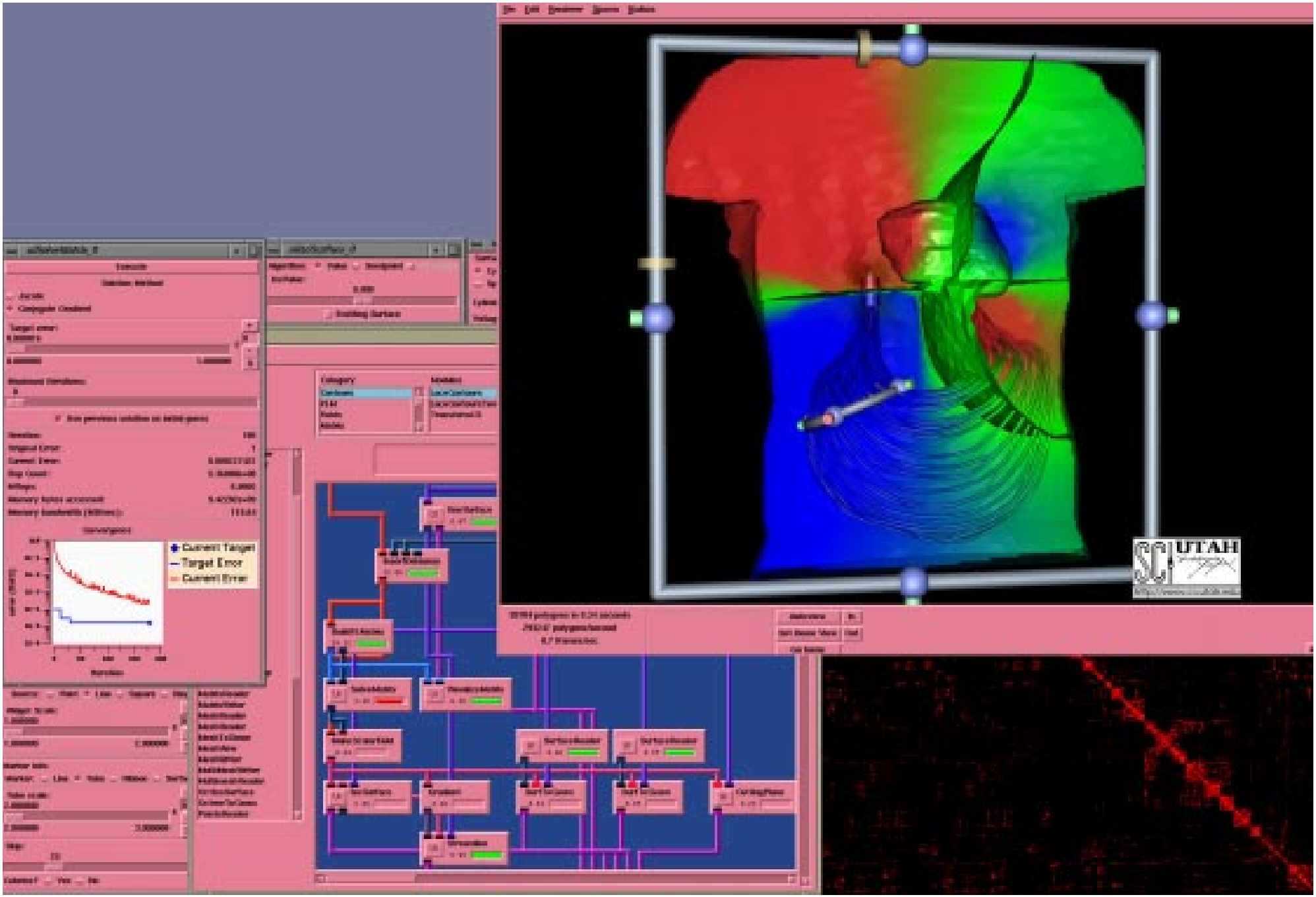
Real distributed applications require coordinating many resources

- **Scale in geographic and organizational scope (see [2])**



♦ Scale in the multiplicity of abstractions needed by different grid user communities

audience	needed services/interfaces
» Lay public, schools » Community emergency services » Military	Web browser / kiosk
Application domain scientists and engineers	Problem Solving Environments / application frameworks
Application domain computational scientists and tool developers	Middleware supporting: • distributed computation • aggregated/federated access to catalogued data • computer mediated collaboration • multiple programming paradigms
Distributed system developers	Job management, access control, generalized communications services, resource discovery and brokering
Middleware / grid common service developers	Local resource managers (queuing, network QoS, scheduled tape marshaling), security services, network services, resource information bases



Problem Solving Environments are the Primary Interface to the Grid for Scientists and

Grid Architecture

The grid may be envisioned as a layered set of basic services and middleware that supports different styles of usage (e.g. different programming paradigms).

However, the implementation is that of a continuum of hierarchically related, independent and interdependent services, each of which performs a specific function, and may rely on other grid services to accomplish its function.

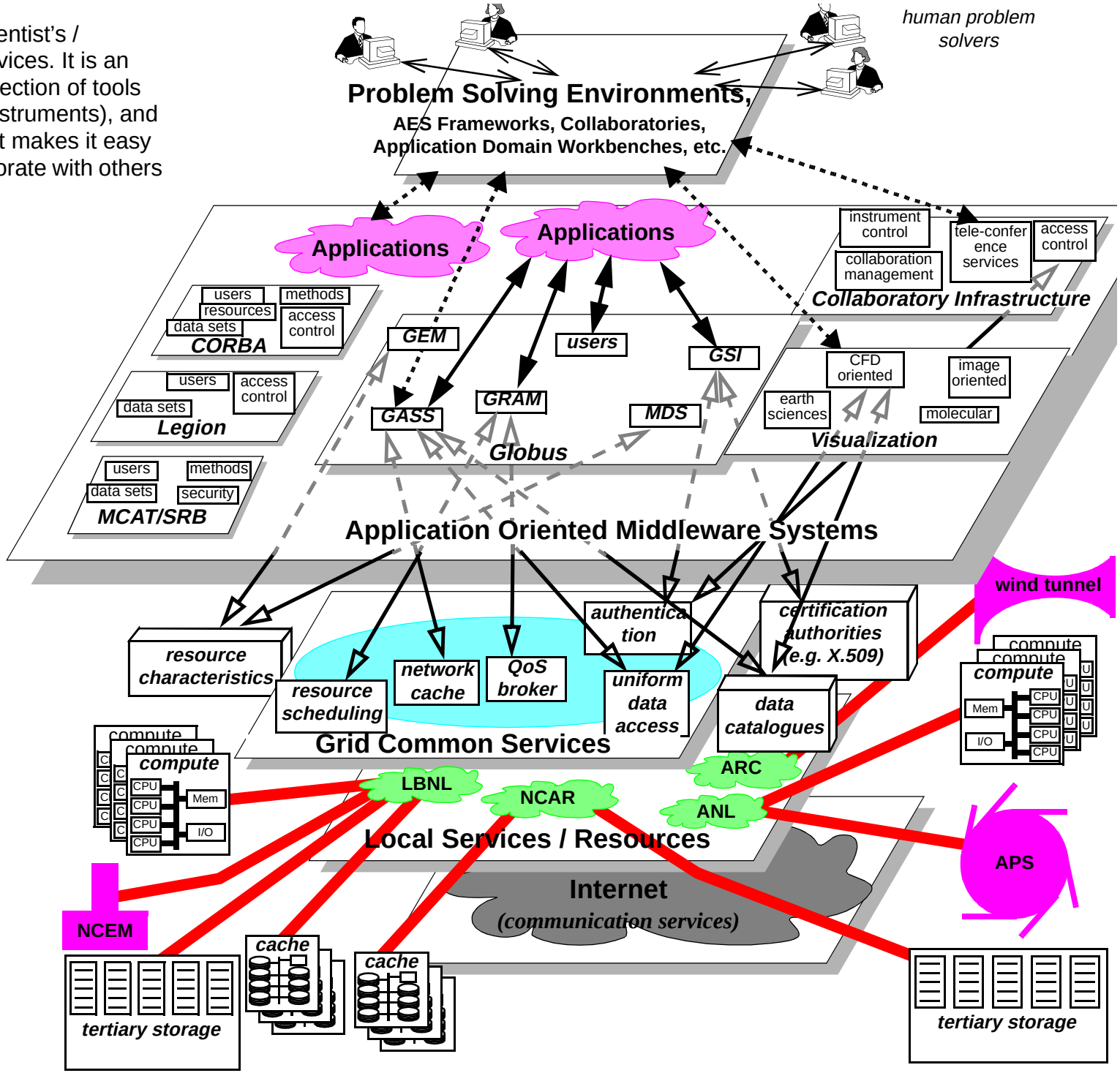
Further, the "layered" model should not obscure the fact that these "layers" are not just APIs, but usually a collection of functions and autonomous management systems that work in concert to provide the "service" at a given "layer."

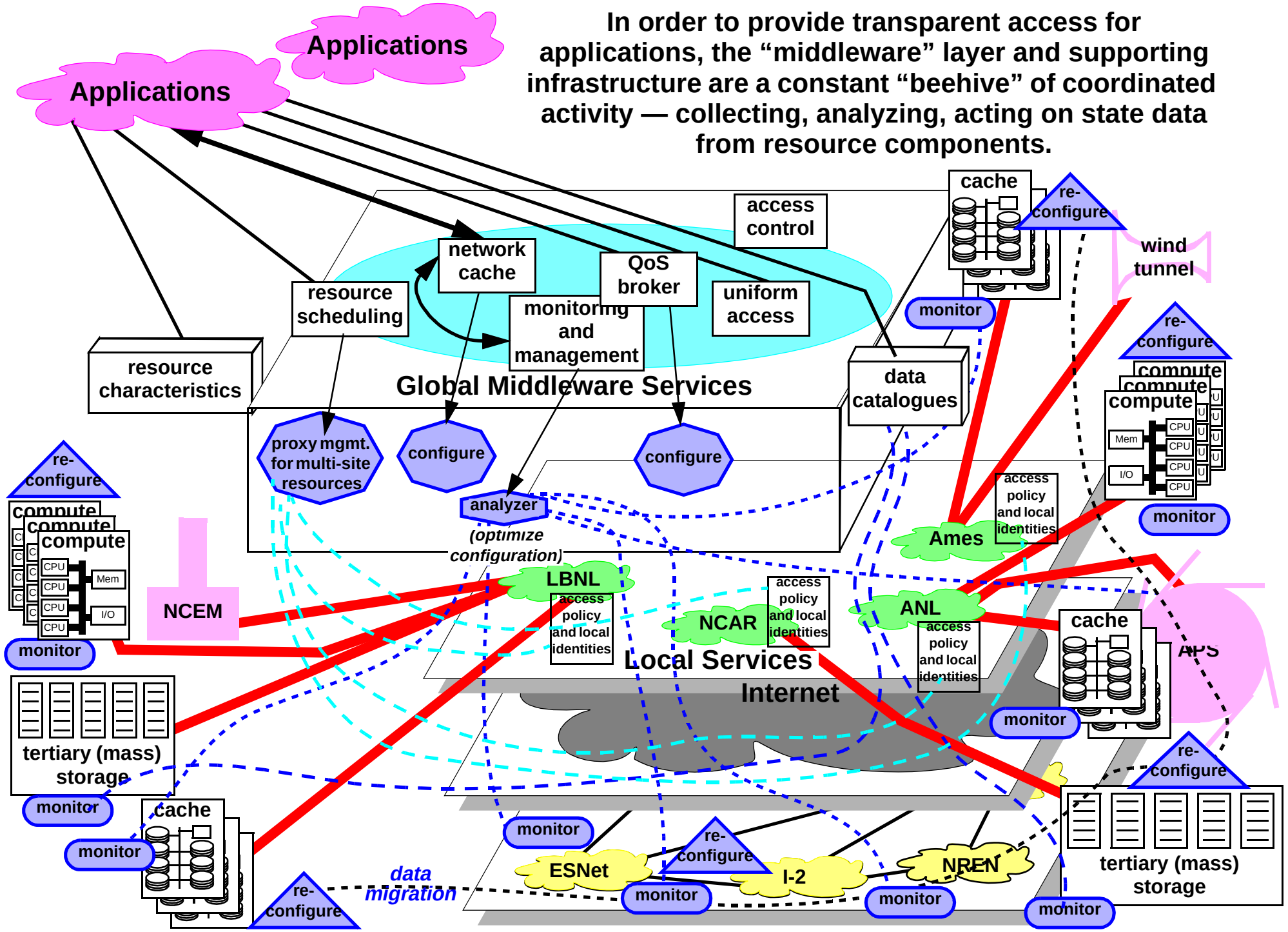
The PSE layer provides the scientist's / engineer's interface to Grid services. It is an application domain-specific collection of tools (e.g. simulations, databases, instruments), and a "workbench" environment that makes it easy to use those tools and to collaborate with others working on the same problem.

The middleware layer provides different styles of service interfaces for application developers to access the basic Grid services.

Grid services are "standard" interfaces for the functions needed to build and manage distributed applications of all sorts.

Most "resources" are "local" and will have their own resource managers and use policies. It is the use mechanisms and interfaces for the local resources that the Grid common services are intended to homogenize.



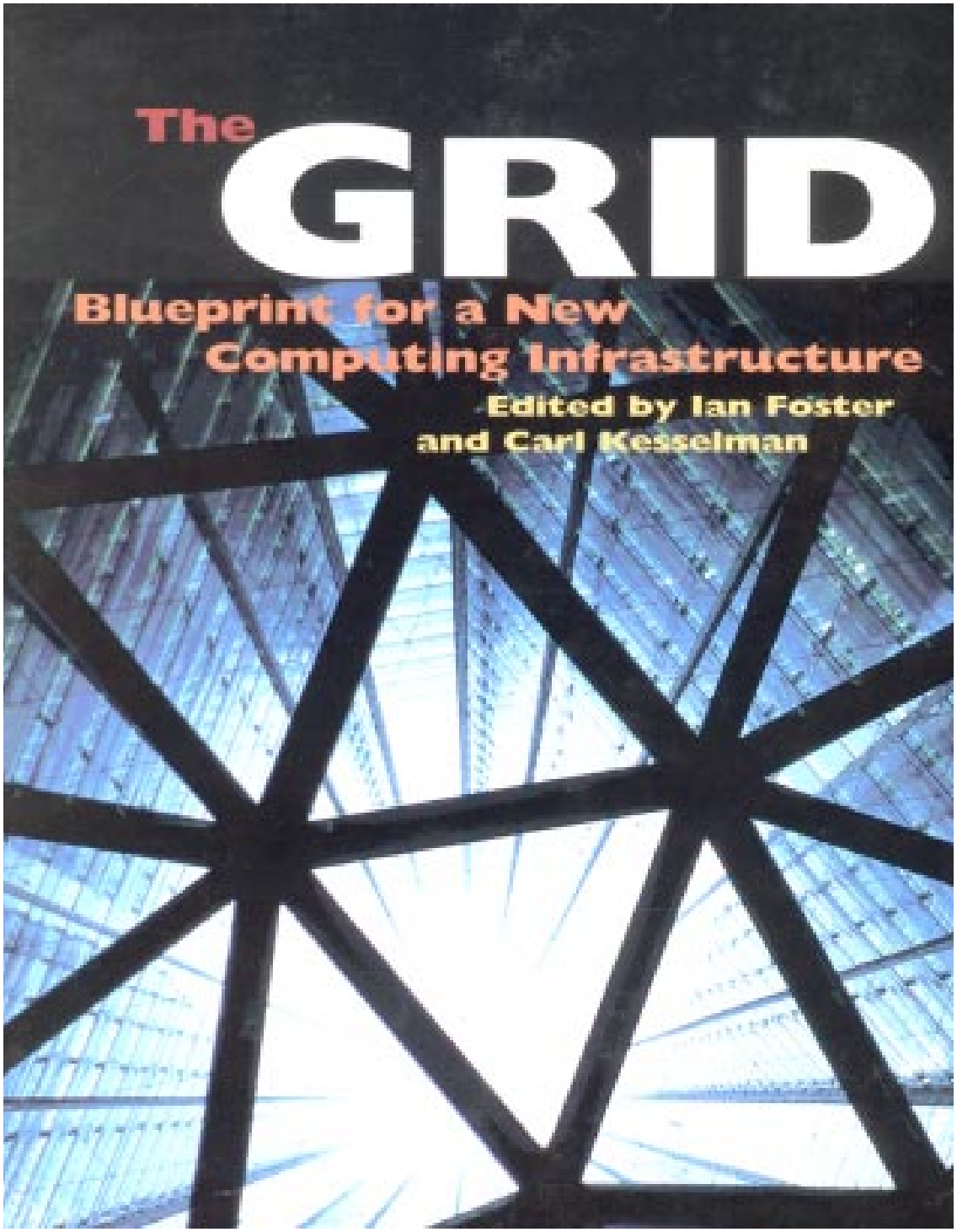

In order to provide transparent access for applications, the "middleware" layer and supporting infrastructure are a constant "beehive" of coordinated activity — collecting, analyzing, acting on state data from resource components.
Applications
Applications
access control
network cache
QoS broker
resource scheduling
monitoring and management
uniform access
resource characteristics
Global Middleware Services
data catalogues
proxy mgmt. for multi-site resources
configure
configure
analyzer
(optimize configuration)
cache
re-configure
wind tunnel
monitor
re-configure
compute
compute
compute
CPU
CPU
CPU
CPU
Mem
I/O
monitor
re-configure
compute
compute
compute
CPU
CPU
CPU
CPU
Mem
I/O
monitor
access policy and local identities
Ames
NCEM
LBNL
access policy and local identities
NCAR
access policy and local identities
ANL
access policy and local identities
Local Services
Internet
cache
APS
monitor
tertiary (mass) storage
monitor
cache
monitor
monitor
re-configure
ESNet
I-2
NREN
monitor
monitor
re-configure
tertiary (mass) storage
monitor
data migration
re-configure

What are "Grids"

- **Comprehensive and consistent set of uniform, easily used, and location independent services that are needed to *routinely* build, operate, and manage *transient and widely distributed* large-scale application systems involving computation, diverse data, real-time instrument systems and collaborative facilities.**

- **Operational infrastructure supporting the services**

The overall goal is to facilitate the solution of large-scale, complex, multi-institutional / multi-disciplinary data and computational based problems of science and engineering.



Together these form environments that are called "grids" [1].

- **The Information Power Grid project is developing and evolving these technologies into a *prototype production* computational and data grid, providing the infrastructure for widely distributed systems.**

Part II: The Information Power Grid Project

Requirements

General capability and services requirements come from experience with the way science and engineering R&D uses computer related resources.

Specific IPG requirements come from analyzing NASA applications:

- **Aerospace Engineering Systems**
- **Earth Sciences, Data Assimilation Office**
- **Astrobiology**

The overall requirement is for an approach that will address a broad spectrum of NASA's computing, data management, and live data source needs.

IPG Overall Goals

- Independent, but consistent, collections of tools and services that support routine construction and use of large-scale, widely distributed applications, data archives, instrument systems, and collaborations.
- Support for “easy” construction of application domain Problem Solving Environments as the primary science and engineering interface to grids.
- Operational grid environment incorporating major computing and data resources at multiple NASA sites in order to provide an infrastructure capable of routinely addressing larger scale, more diverse, and more transient problems than is possible today.

Approach: Grid Services

- Toolkits for:
  - Constructing application "frameworks" / Problem Solving Environments
  - Integrating computer mediated, distributed human collaboration
  - Integrating laboratory / experiment / analytical instrument systems
  - Grid enabled visualization and data exploration
- "Global shell" - having the entire grid environment appear like the IMac / Windows / X workstation on the user's desk - this is "basic PSE" that provides "seamless" job control and resource access

- **Grid common runtime services**
  - **Resource discovery and brokering**
    - **find an object (e.g. data base, a computation, a server) with a given set of properties**
    - **install a new object/service into the grid**
    - **make that object known as a grid service**
  - **Co-allocation for a complex mix of resources**
    - **reservation and QoS for all services (e.g. CPU and network bandwidth)**
  - **Global queue management for compute intensive tasks**
  - **Generalized fault management**

- **Uniform interfaces to multiple “annotated” data archives**
- **Data location management**
  - **local, remote, cached (where?)**
  - **remote I/O for large data sets**
- **Authentication and security**
  - **access to the grid and its resources are based on a single cryptographic credential maintained in the users desktop / PSE environment(s)**
- **Authorization and access control**
  - **maintenance of stakeholder rights**

- **Uniform naming and location transparent access to resources such as data objects, computations, instruments and networks**
  - **urn's that work through Grid-wide object brokers**

♦ The Programming Environment

- **Support for multiple programming styles* in multi-platform, heterogeneous computing systems**
- **Distributed debugging and performance monitoring**
- **Grid-enabled visualization and data exploration**
- **Uniform model for data access****
- **Grid communication libraries**
- **Support for CORBA, Enterprise Java Beans, DCOM**

---

*Current examples include threads and shared memory, Fortran plus MPI (for message passing in parallel systems), C++ coupled to object oriented databases (for structured data storage and management), CORBA (being used to facilitate construction of composable systems - “reusable modules”), and general object oriented approaches such as Java/Jini and Legion are being experimented with.

**What does a uniform access interface look like that incorporates MPI/IO files, legacy data sets in deep archives, Web based documents, and instrument data streams? (E.g. Vanderbilt’s work.)

- **Services essential for scalability**
  - **Brokering* and autonomous management**
  - **Access to "global" system state information**
  - **Policy based access control and use-condition management, and security for all resources**

*"Brokering" = given a description of the resources needed to solve a problem, locate candidates in the grid and negotiate for their use.

♦ Operability

- A “grid common information base” that provides global information about the configuration and state of the grid
- Diagnostic tools so opns/systems staff can investigate remote problems
- Tools and common interfaces for system and user administration, auditing, accounting, etc.
- Benchmarks/regression analysis tools for performance, reliability, and sensitivity testing

Part III: Implementing IPG

- We are not starting from scratch: There is a sizable academic/government/commercial R&D computer science community working toward grid computing environments.
- Ames' NAS Division is restructuring its organization and computing approach around IPG
- Engineering is well under way for building IPG testbeds, including the first prototype-production grid system that will routinely provide distributed computation and transparent tertiary storage access for a sizable user community in wide area networks.

- **Information Power Grid is a sizable collaboration led by NAS/ARC, and directly involving:**
  - **GRC (LeRC), LaRC, and GSFC**
  - **the NSF high performance computing consortia: Alliance (NCSA) and NPACI (SDSC)**
  - **additional university partners**
  - **industrial partners (e.g. Cisco, SGI, Sun)**
- **A detailed implementation plan is being developed that includes the following top-level elements:**

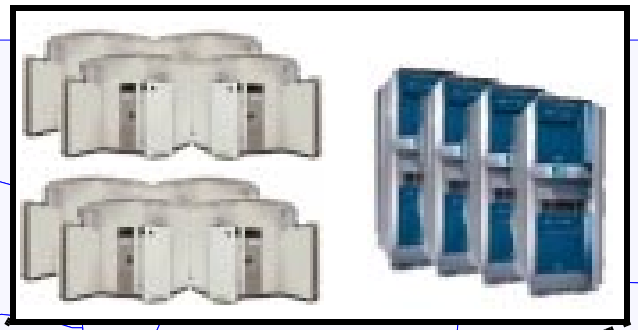
Two year goals:

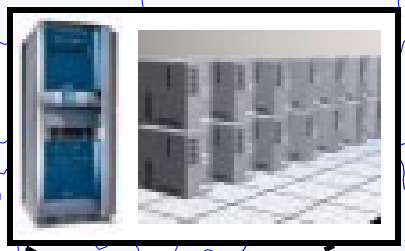
- **An operational IPG "prototype-production," heterogeneous, distributed environment that provides access to computing, data, and instrument resources at several NASA Centers**
- **Support for multiple programming environments (Globus, Legion, CORBA, Java/Jini)**
- **At least two significant application classes using unique characteristics of IPG (e.g. co-scheduling, location transparent data management, automated fault management)**

- **A prototype collaborative workbench supporting multi-disciplinary science and engineering (e.g. an Aerospace Engineering Systems framework and an Astrobiology PSE)**
- **Support for real-time access to scientific and engineering instrumentation systems**
- **Integration with CosMO**
- **Definition of a long-term R&D program addressing grid computing, information, real-time instrument, and collaboration issues**

Current Progress

- Grid common information base
- Globus [2] as the initial grid “runtime” system
- Global queue management design and implementation
- CPU and bandwidth reservation
- SRB/MCAT for uniform MSS access
- X.509 certification authority for IPG user identity
- Operations and user services infrastructure
- Programming and application services
- IPG Testbed





Boeing

ARC

JPL

SDSC

GRC

CMU

GSFC

LeRC

NREN

IPG initial QoS Testbed

NTON-II/Supernet NGI testbed

Initial Configuration of IPG CX and DX Testbeds

IPG Next Generation Internet Projects

High-speed, wide area networks are an essential and inseparable aspect of grids.

- **IPG will enable the applications envisioned for NGI by providing the distributed systems infrastructure.**
- **NGI will enable IPG through R&D and testbeds for several key network communications technologies.**

♦ IPG NGI projects:

- Very high data-rate applications that will use IPG as infrastructure, which, in turn, uses NGI for network capacity:
  - Based on OC-48 (2.5 Gbit/s)

    JPL ↔ ARC ↔ Boeing, Seattle, potential projects include:
    - aviation safety “help desk”
    - remote operation of wind tunnels
    - high data-rate access to historical wind tunnel data
    - image processing and SAR processing with JPL
    - remote VR
- Very high data-rate and high volume wide area data management

IPG Two Year Grid System Milestones

1) Globus as the initial runtime environment providing a robust, usable, widely deployed, distributed computing framework:

- **resource management**
  (standardized interface to various local resource management systems (GRAM/RSL) and automated allocation of collections of resources (DUROC))
- **remote data access**
  (automated staging and transparent remote access to files (GASS and RIO))
- **execution environment management**
  (executable code and library staging (GEM))
- **security**
  (single sign-on, authentication, authorization, and optional confidentiality within IPG system)

- **monitoring and fault detection**
  (services supporting fault detection and recovery into Globus applications (HBM, et al))
- **system/resource information infrastructure**
  (Global access to information about the state and configuration of all system components (MDS))
- **Grid programming services**
  (support for writing parallel-distributed programs (Nexus, MPICH-G))
- **Grid administration**
  (tools for the Grid operators)

2) Grid runtime environment II: Augmented services

- **a global shell**
  (support for managing various task models, workflow mgmt., signals, I/O mgmt., etc.)
- **facilities for time based scheduling (reservation) of resources, requirements matching, and QoS**
  (support for co-scheduling, relative priority, resource profile queries, etc., for all

resources)

- policy based access control services
- application performance monitoring
- distributed debuggers
- services supporting CORBA, Legion, Java (and DCOM?)
- support for commodity platforms
  (cycle stealing and cluster scheduling for both Unix and NT)
- global queue management

3) Advanced services integrated into IPG:

- high-performance, high-capacity, metadata based digital data libraries for transparent management of multiple archival storage systems
- grid enabled visualization, “steering,” and data exploration tools

- **autonomous agent framework providing support functions for both humans and resources (autonomous infrastructure management)**
- **services and tools supporting knowledge based, generalized fault management services and tools**
- **identification and implementation of the services needed to support building collaborative PSE/workbench systems**

4) Integration of instrument systems

- **application caching, hard/soft resource reservation, reliable multicast, multi-operator control management**

Two Year Operational Milestones

1) Persistent, "large-scale," heterogeneous, distributed testbed that enables applications that cannot be done today

- **significant CPU resources**
  ($\geq$ 50% of all non-COSMO, NAS resources + IPG systems at LeRC, LaRC, GSFC, and ICASE ($\cong$ 30-40 Gflop) + resources from Alliance and NPACI)
- **stable and predictable application environment**
  (long term resource usage policies + operational support)
- **widely distributed**
  (usable resources at multiple sites)
- **running "stable" Globus "beta" release**
  (new services will be incorporated into the testbed, which will have the direct involvement of the Globus developers)

- **procedures for “bullet proofing” and validating the beta releases prior to installation in the testbed**
  (developers testbed: clean up, validate, and integrate into the beta distribution)
- **user documentation and consulting**
  (users will both be supported and be partners in the IPG evolution)
- **operations and system admin. services and documentation**
  (testbed will be integrated into the NAS 7X24 production operations environment)

2) Comprehensive benchmark suite

- **develop performance measurement tools for the IPG**
- **“regression” suite for monitoring stability of IPG functionality and performance during software upgrades**
- **provide programming examples**

3) Multi-disciplinary prototype applications

- requirements analysis and support for an integrated AES application (e.g. airframe and propulsion)
- EOS/DAO
- Astrobiology projects
- RLV parameter studies (?)

4) Operating in COSMO environment

- establish COSMO criteria and mechanism for transition to production

5) An operating “grid” software “standards” process involving computing and aerospace industry, government agencies, and universities

6) Advanced heterogeneous computing systems testbed environment

- maintain part of IPG as a viable testbed for prototyping advanced distributed systems and prototype applications
- use IPG testbed to evaluate and promote advanced computing platforms by porting the distributed systems software to new platforms and incorporating them into IPG

Two Year R&D Objectives

1) Distributed-parallel algorithms in the grid environment

2) Ultra high-speed distributed systems

- Network R&D
- Infrastructure design, monitoring, and testing
- Platform and I/O subsystems
- Data management systems
- Applications architectures and algorithms
- High data rate instrument systems
- Proto-applications

3) Advanced program execution environment

- Network and resource-aware adaptation

Technology Refresh

An on going task in IPG will be participation in developing, acquiring, deploying, and evaluating new computing and storage resources.

1.0 Computing elements

1.1 New Computing Architectures

HTMT is one example of an advanced, high-end computing architecture.

Major advances in key technologies present the opportunity to achieve petaflops scale computing within a decade -- far less time than anticipated through the evolution of conventional semiconductor technology. Superconductor Rapid Single Flux Quantum logic can operate in the region of 100 GHz with power consumption within the cryostat of less than 50 watts for a Petaflops. Holographic photorefractive systems may be able to store up to 100 Gbits in a cubic centimeter and deliver bandwidths approaching a terabit per second; with advanced spectral

Hybrid Technology Multi-Threaded Architecture

Aggressive Latency Management Through Multi-Level Multi-Threading

Challenge:

- **Achieve high sustained performance in the presence of disparate cycle times and latencies**
- **Exploit superior "operating point" characteristics enabled by multi technology implementation**
- **Provide a uniform execution framework for generality and programmability**

Concept:

- Very high speed processor logic combined with deep memory hierarchy for very high density storage
- Multi-Level multi-threading
  - Processors employ single cycle thread switching
  - Smart memories "percolate" coarse grained contexts up to processors

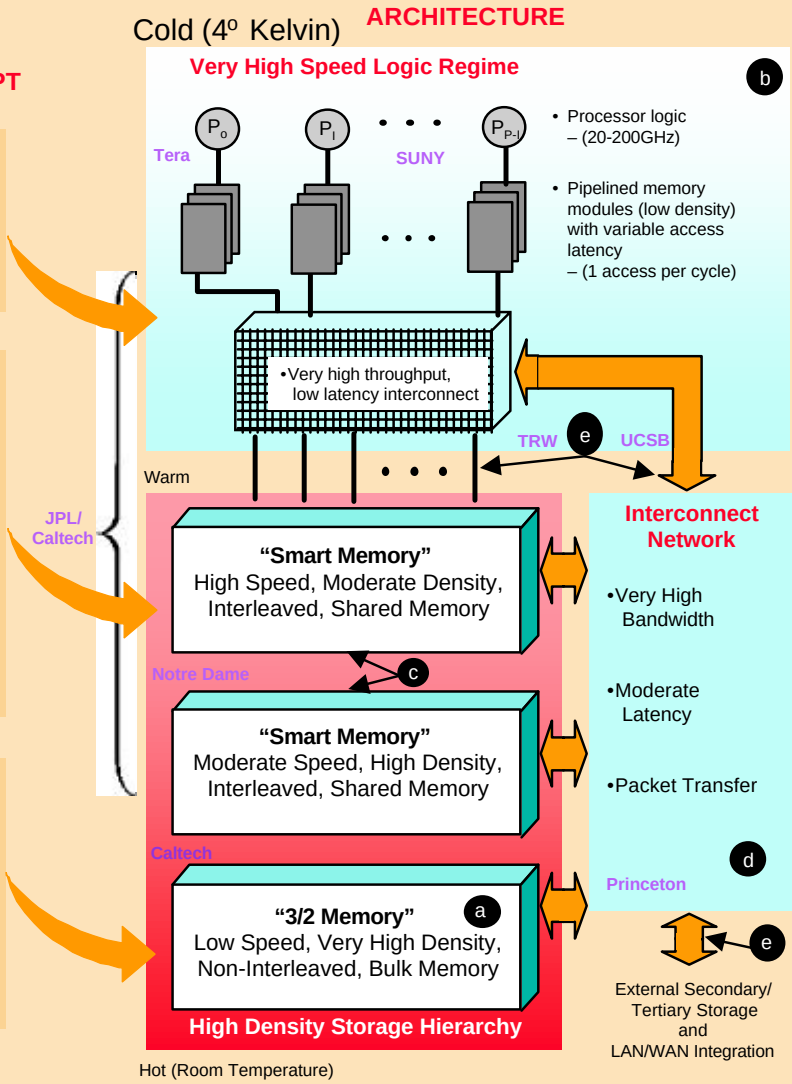
LATENCY HIDING CONCEPT

Top level single cycle thread switching hides 100s of cycles of latency across processor array and to top of memory hierarchy

Second level "context percolation" multi-threading synchronizes and manipulates coarse grained threads (processes). Ready contexts are migrated to the flattop of the smart memory hierarchy while pending contexts are migrated down to high density lower levels of smart memory hierarchy

"3/2 memory" sits between primary memory and secondary storage. Provides very high bandwidth high density storage but at x 100 longer access latency than main memory. Employs data streaming for latency hiding

ARCHITECTURE



EXAMPLES OF EMERGING TECHNOLOGIES

a — Holographic Storage:
- Very high density storage
  – Approaches tera bit/cm^3
- Very high bandwidth
  – 100-1000 G bits/sec
- Very low power storage employs photo refractive or spectral hole burning

b — Superconductor RSFQ Logic:
- Very high speed
  – 20-200 GHz
- Very low power
  – Microwatts per gate
- Moderate density
  – 1.5 m (present)
  – 0.5 m (possible)

c — Processor in Memory (PIM):
- Merges logic and memory on single chip
- Exposes high intrinsic memory bandwidth
- Reduces power per operation
- Enables smart memory support of multi-level multi-threading

d — Multi-Level Minimum Logic Optical Network:
- Very high bandwidth fiber optics
  – 100-250 Gbps per fiber
- Very high degree packet switched network
  – ~100,000 ports
- Minimum logic optical nodes
  – 2 in, 2 out, 3 gate pure optic data path
- Local flow control – no global synchronization
  – Low latency 10-30 ns

e — III-V Semiconductor Logic:
- In GaAs P device technology
- Very high speed switching
  – 10-100 GHz
- Low density, high power
- Interface electronics

TEAM: JPL, CALTECH, SUNY, PRINCETON, NOTRE DAME, UNIVERSITY OF DELAWARE, TERA COMPUTER, TRW, UCSB

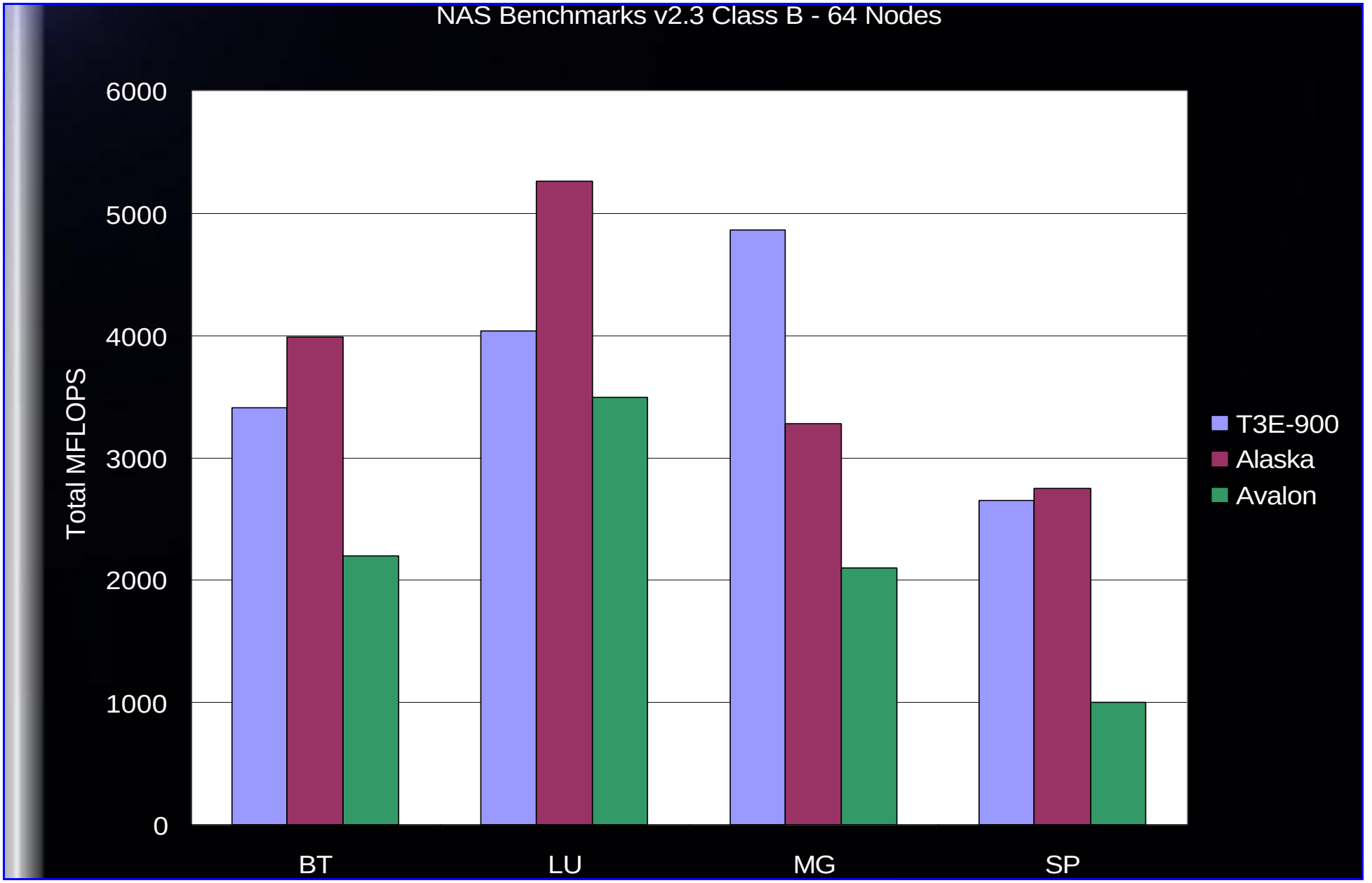
Sponsored by: DARPA, NSA & NASA

hole burning techniques, these attributes may increase by a factor of 100. A new packet switching optical network that incorporates a multi-level structure with minimum logic per optical switching node will provide interconnection among tens of thousands of ports with latencies of between 10 and 30 nanoseconds using only localized flow control. PIM technology permits fully integrated memory and logic cells on a single chip, exposing the full memory bandwidth of the internal row buffers and significantly reducing package and pin counts.

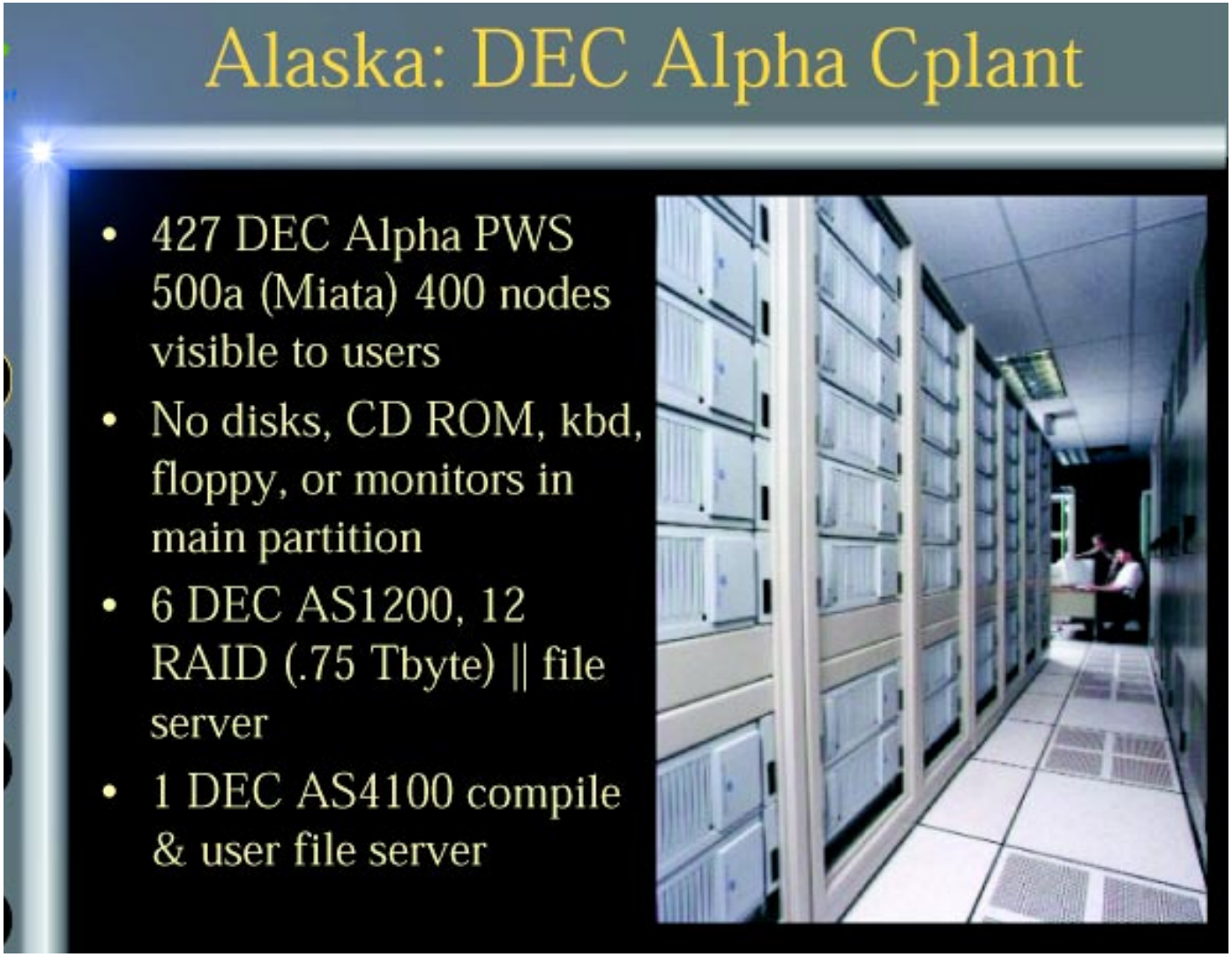
This study effort investigates a computer architecture that, combined with a hybrid technology strategy, exploits these new technologies to achieve revolutionary performance. However, the large disparity in access latencies between the fastest superconductor logic and the slowest optical memory present a formidable challenge to efficient system use. Current multithreaded architecture techniques hide latency by context-switching among concurrent threads and only activating instructions whose operands are available. While these methods may be effective at hiding latencies in the range of 100 to 1000 cycles, a hybrid technology architecture will impose latencies as much as a thousand times greater. The hybrid technology architecture features new multi-threading techniques to manage the much greater latencies that result from the combination of the new technologies.

From: "A Hybrid Technology Multithreaded Computer Architecture for Petaflops Computing" by Thomas Sterling http://htmt.cacr.caltech.edu/Overview.html

1.2 “Modular” supercomputers

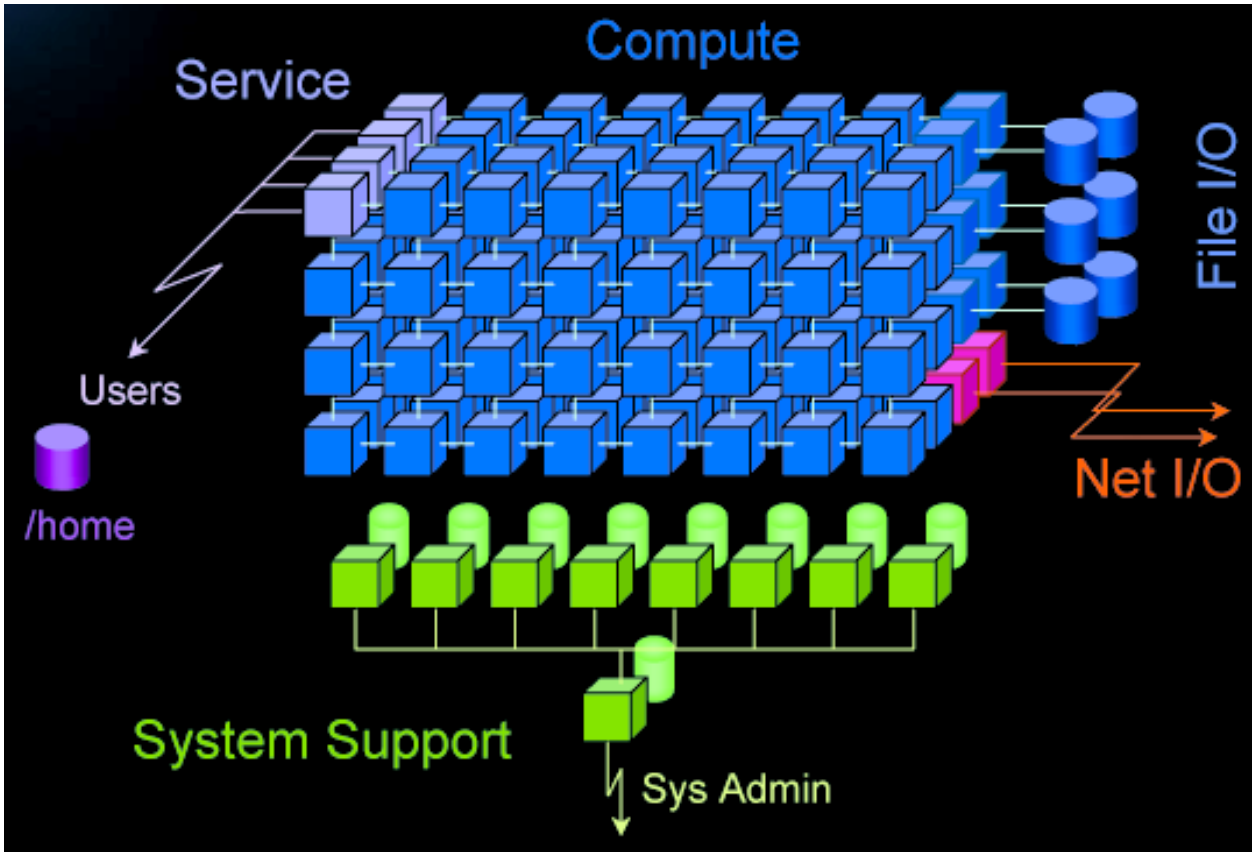


Sandia’s “Alaska” Cplant Alpha PC cluster performs like a supercomputer.
(see See http://www.cs.sandia.gov/cplant)



The Computational Plant project at Sandia National Laboratories is developing a large-scale, massively parallel computing resource from a cluster of commodity computing and networking components. We are combining the knowledge and research of previous and ongoing commodity cluster projects with our expertise in designing, developing, using, and maintaining large-scale MPP machines. Our goal is to provide a commodity-based, large-scale computing resource that meets the level of compute performance needed by Sandia's critical applications.

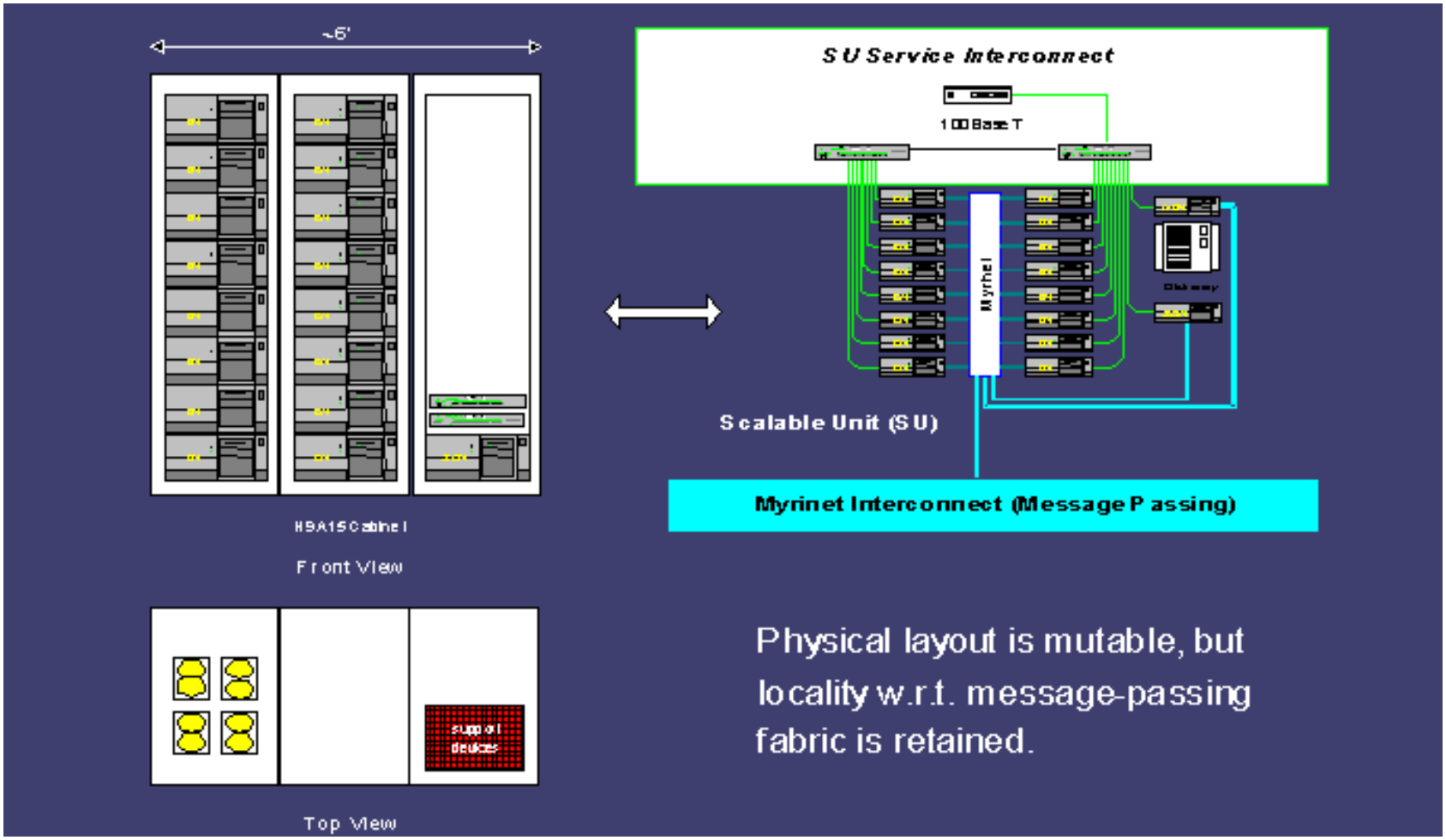
Sandia National Laboratory's "Computational Plant"



Conceptually how the system is divided into distinct partitions, each with its own purpose. The service partition is where users log in and launch their parallel applications. There is always at least one service node. If the machine needs to support many simultaneous users, the service partition can grow. Applications have access to user file system from the compute nodes they are running on. These user files are made visible through the high-performance network, using Portals, not NFS.

The compute partition is where the processes of a parallel application execute. Users cannot log in directly to a compute node. Our compute nodes do not have local storage. Swapping is turned off, and files go through the loader to the user's home file system, or in parallel to our striped mass storage server.

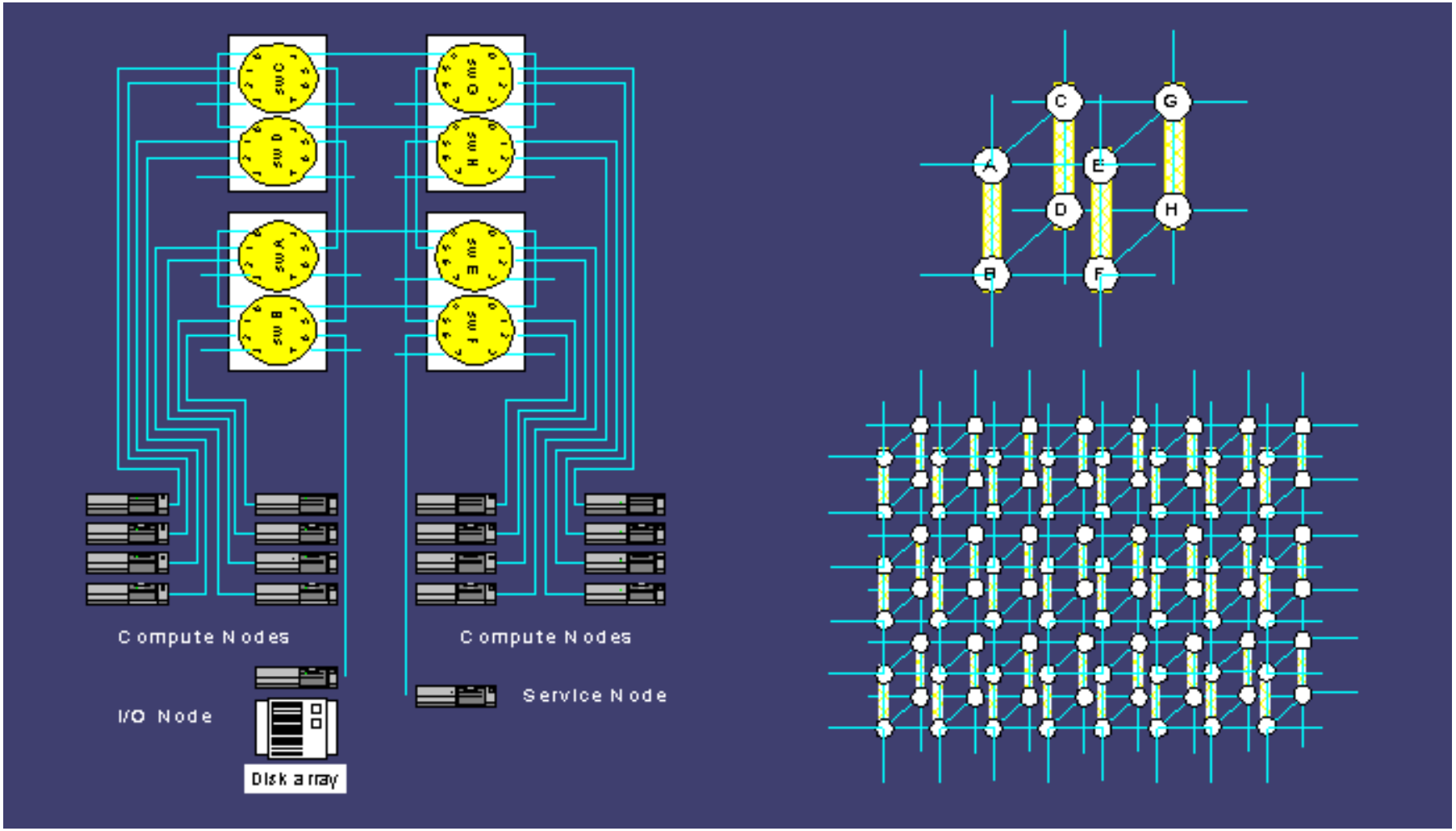
The I/O partitions (file and network) are nodes with additional hardware and services. They can be accessed from the compute (and service) partition through the high-speed local network (Myrinet). The system support partition is not directly visible by users of the system. It is not connected to the Myrinet, but provides system software and control (boot, console, power, reset) to all the nodes for system administrators.



"Scalable Unit = Building Block"

In Cplant, grow and prune is done by using scalable units (SU). Each SU contains a number of nodes, a system support station (SSS), a high-speed network fabric with connections to other SUs, and all the hardware necessary to maintain the nodes: remote power control, bootp, and serial console lines for diagnostics and system administration. The nodes can be configured to be compute nodes, I/O nodes (if they have disks or WAN networks attached), or service nodes. Not every SU has to have a service node, but there has to be at least one somewhere in the system. The service node(s) is where users log in and launch parallel applications.

The current implementation of an SU consists of 16 nodes. They are DEC Alpha PWS 500a (Miata) (500MHz, 192MB RAM, 2MB L3 cache). They do not contain a hard disk, CD-ROM, keyboard, mouse, or a video adapter.



"SAN Configuration"

The high-speed network that is in use on the DEC Alpha Cplant, is a Myrinet. The current topology is a modified hypercube. There is a 16-port switch (8 SAN + 8 LAN connectors) in each cabinet. I.e. there are two 16-port switches per SU for a total of 50 switches. We use the 32-bit PCI bus LANai 4.3 interface cards with 1MB of local RAM.

The diameter of the network is 4 switch crossings, with a 2.5 hops average. The bisection bandwidth (peak) of the network is more than 10 GB/s bi-directional. The main limitations to highest possible bandwidth and low latency are limits of the PCI bus and software overhead.

2.0 Mass storage

Optical holographic storage systems are likely to have a major impact on archival storage strategies in the next five years.

Holographic data storage holds the promise of combining very high-density storage of data with high data rates. This potential has been well understood for several decades and stimulated extensive research efforts in the past. Nevertheless no viable commercial products are currently available. After a brief introduction into the concepts of holographic data storage recent changes in the environment, that have stimulated a flurry of activities in this field, are discussed. Then the current status of research in this field is being reviewed and the open issues are highlighted.

Hans J. Coufal, IBM Research Division, Almaden Research Center

IPG Four Year Objectives

1) A high performance, widely distributed, global file system

- provide a completely uniform view of the Grid application environment regardless of location

2) Transparent fault tolerance and reconfiguration

- the Grid will dynamically reconfigure itself and provide sufficient assistance and information to running applications that they can also reconfigure

3) Numerical techniques optimized for parallel-distributed environments

- significant progress in developing new numerical algorithms optimized for the Grid environment

4) Infrastructure security

- operational security for the communications, systems, and other resources of the Grid

5) Distributed object programming paradigm (probably Java based) integrated into the Grid framework

6) Facilities for coupled computational simulation and experiments, and computational steering

7) High capacity, high performance, prototype-production metadata based data repositories that support data management, mining, and exploration for AES and DAO

8) Prototype ASE Framework components

- collaborative workbench building environments that enable coordinated, multiple component model operation

IPG Six Year Objectives

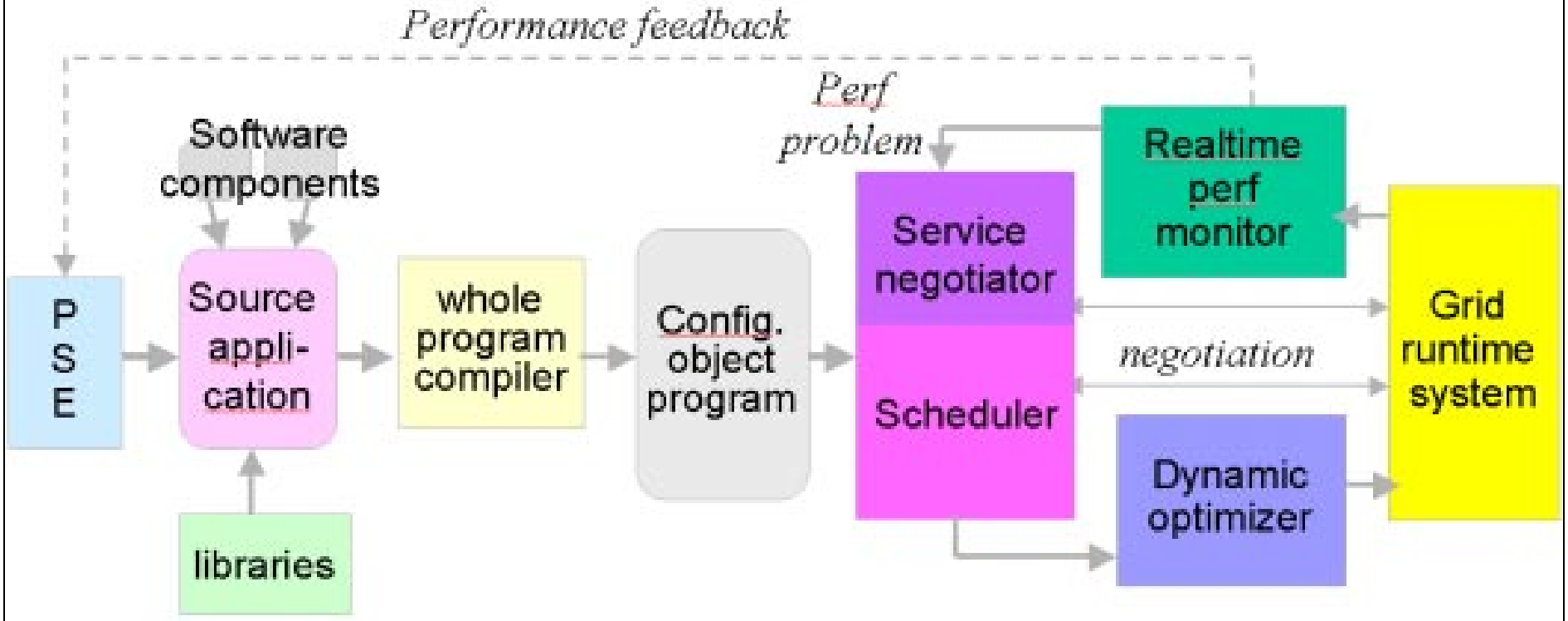
1) Advanced programming paradigms including adaptive compilation and resource driven dynamic linking

2) Application performance model based resource scheduling

These objectives are closely aligned with, and will be coordinated with NSF's Next Generation Software program:

"The NGS program fosters multidisciplinary software research under two components: Technology for Performance Engineered Systems (TPES), and Complex Application Design and Support Systems (CADSS). The overall thrust of NGS will be research and development for new software technologies integrated across the systems' architectural layers, and supporting the design and the operation cycle of applications, computing and communications systems, and delivering quality of service (QoS). The TPES component will support research for methods and tools leading to the development of performance frameworks for modeling, measurement, analysis, evaluation and prediction of performance of complex computing and communications

- **Grid-aware Programming**
  - development of adaptive poly-applications
  - integration of schedulers, PSEs and other tools



(Berman, Darema, Gannon, Kennedy, et al.)

systems, and of the applications executing on such systems. The CADSS component will support research on novel software for the development and run-time support of complex applications executing on complex computing platforms; CADSS fostered technology breaks down traditional barriers in existing software components in the application development, support and runtime layers, and will leverage TPES developed technology for delivering QoS.
See http://www.nsf.gov/cgi-bin/getpub?nsf998

3) Prototype ASE Framework

- coordinated operation of multiple AES component models in a framework that represents and can (partially) evaluate an operational vehicle

IPG is putting significant resources into both R&D, and testbed construction and operation, in about equal proportion.

R&D will be an on-going IPG activity in order to ensure that the grid continuously evolves through incorporating leading-edge technology.

Areas of New R&D for Data Intensive Applications

- Services for locating the data resources needed to solve a given problem
- Integration of tertiary storage systems with digital libraries / metadata catalogue systems in widely distributed environments
- Widely distributed storage architectures that incorporate resource and bandwidth QoS
- Integration of policy based access control
- Integration of data location management with uniform data access techniques
- Fault tolerant distributed storage and catalogue systems
- Development of automated cataloguing techniques and self-describing data semantics from metadata

References and Notes

[1] *The Grid: Blueprint for a New Computing Infrastructure*, edited by Ian Foster and Carl Kesselman. Morgan Kaufmann, Pub. August 1998. ISBN 1-55860-475-8. http://www.mkp.com/books_catalog/1-55860-475-8.asp

[2] Globus is a R&D grid system that is the starting point for IPG. Globus is described at www.globus.org

[3] "Real-Time Generation and Cataloguing of Large Data-Objects in Widely Distributed Environments," W.Johnston, Jin G., C. Larsen, J. Lee, G. Hoo, M. Thompson, and B. Tierney (LBNL) and J. Terdiman (Kaiser Permanente Division of Research). Invited paper, International Journal of Digital Libraries - Special Issue on "Digital Libraries in Medicine". May, 1998. http://www-itg.lbl.gov/WALDO/

[4] DPSS: The Distributed-Parallel Storage System (DPSS) is a scalable, high-performance, distributed-parallel data storage system developed in the MAGIC Testbed. The DPSS is a collection of wide area distributed disk servers which operate in parallel to provide logical block level access to large data sets. Operated primarily as a network-based cache, the architecture supports cooperation among independently owned resources to provide fast, large-scale, on-demand storage to support data handling, simulation, and computation in a high-speed wide-area network-based internetworked environment. See http://www-didc.lbl.gov/DPSS.

[5] Clipper: The goal of the Clipper project is software systems and testbed environments that result in a collection of independent but architecturally consistent service components that will enhance the ability of applications and systems to construct and use widely distributed, high-performance data and computing infrastructure. Such middleware should support

high-speed access and integrated views for multiple data archives; resource discovery and automated brokering; comprehensive real-time monitoring and performance trend analysis of the networked subsystems, including the storage, computing, and middleware components, and; flexible and distributed management of access control and policy enforcement for multi-administrative domain resources. See http://www-itg.lbl.gov/~johnston/Clipper

[6] MAGIC: "The MAGIC Gigabit Network." See: http://www.magic.net

[7] SCIRun

SCIRun is a scientific programming environment that allows the interactive construction, debugging and steering of large-scale scientific computations. SCIRun can be used for interactively:

- ***Changing 2D and 3D geometry models (meshes).***
- ***Controlling and changing numerical simulation methods and parameters.***
- ***Performing scalar and vector field visualization.***

SCIRun uses a visual programming dataflow system. SCIRun is extensible to a variety of applications and will work with third party modules written in Fortran, C, and C++.

http://www.cs.utah.edu/~sci/software/

[8] NREN network: http://www.nren.nasa.gov/eng/top.html

[9] Supernet: see "Architecture and Engineering" at http://www.ngi-supernet.org/ and http://www.ngi-supernet.org/supernet-backbone.gif

http://nas.nasa.gov/~wej/home/IPG